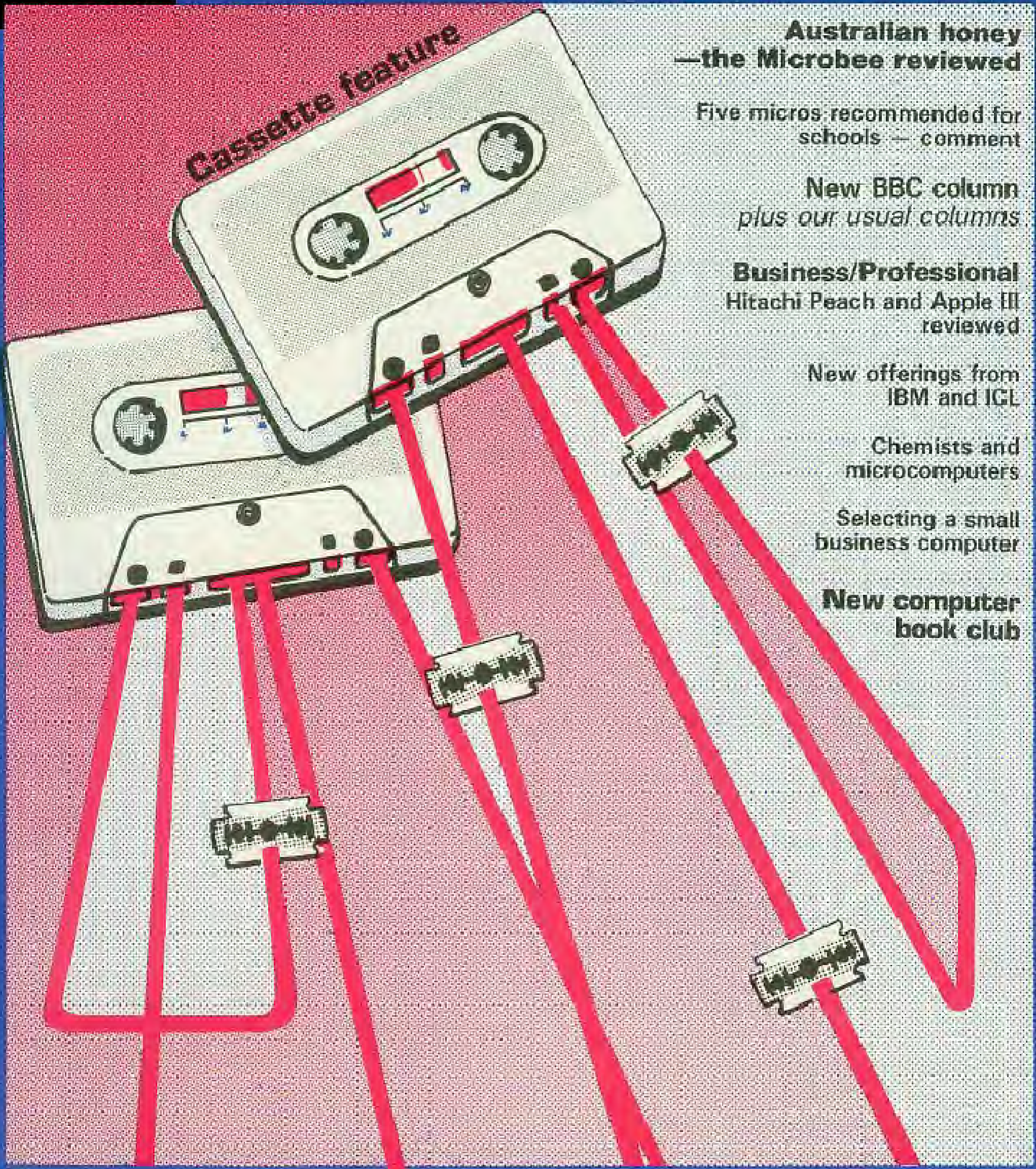
[redacted]AND'S PERSONAL COMPUTER MAGAZINE

[redacted]TS & BYTES

Issue No. 6, March 1983: $1.00



Australian honey —the Microbee reviewed

Five micros recommended for schools — comment

New BBC column
plus our usual columns

Business/Professional
Hitachi Peach and Apple III reviewed

New offerings from IBM and ICL

Chemists and microcomputers

Selecting a small business computer

New computer book club

Commodore COMPUTER and BITS..&...BYTES..

NEW ZEALAND'S PERSONAL COMPUTER MAGAZINE

INVITE YOU TO WIN A....

COMMODORE VIC 20 COMPUTER–FOR THE WHOLE FAMILY!

See last issue for details and

Pick up your entry form from these participating VIC 20 dealers

NORTHLAND
Gemini Home Ltd
30–40 Rathbone Street
WHANGAREI
Phone 84-999

Mid-Northern Services Ltd
No 1 State Highway
OHAEAWAI
Phone 7530

AUCKLAND
Adair Home Appliances Ltd
3 Birkenhead Ave, Birkenhead
AUCKLAND
Phone 439-009

Anderson Electrical Ltd
995 Dominion Rd, Mt Roskill
AUCKLAND
Phone 697-203

Armed Forces Canteen Council
Military Camp, Papakura
AUCKLAND
Phone 299-5487

Ashford Television Ltd
108 Kepa Rd, Orakei
AUCKLAND
Phone 583-203

The Byter Shop
1 Fort Street
AUCKLAND
Phone 32-860

Computer World
Cnr Lorne & Victoria Streets
AUCKLAND
Phone 31-394

Farmers Trading Company Ltd
Hobson St
AUCKLAND
Phone 799-229

K Road Video & Computer Co
65 Pitt Street, Newton
AUCKLAND
Phone 369-655

Micromax
P.O. Box 33968
TAKAPUNA
Phone 479-6024

Organ World of Music Ltd
Como St Entrance — Shore City
AUCKLAND 9
Phone 494-472

Public Service Investment Store
Parkway Arc, 54 Hurstmere Rd
AUCKLAND
Phone 483-157

Rogers Home Appliances Ltd
235 Broadway, Newmarket
AUCKLAND
Phone 549-804

Selcom Electronics
2a Basin View Lane, Parnell
AUCKLAND
Phone 577-109

Stereo & Video TV Services
144 Glenmore Drive, Titirangi
AUCKLAND
Phone 489-979

Sound Centre Ltd
Civic Centre Arc, Browns Bay
& Huron St, Takapuna
AUCKLAND
Phone 479-5265

Soundwork Electronics Ltd
100 Anzac Avenue
AUCKLAND
Phone 798-075

Superior Electronics
430 Mt Eden Road
AUCKLAND
Phone 605-218

Tivetel Ltd
88 Galway Street, Onehunga
AUCKLAND
Phone 660-505

Mr R. Thornton
8 Portage Road, Papatoetoe
AUCKLAND
Phone 278-3185

John Walker Music Ltd
60 O'Shannessey St, Papakura
AUCKLAND
Phone 299-8827

WAIKATO/KING COUNTRY
Busy Bee Computing Ltd
Cnr Mannering & Gresham Pl
TOKOROA
Phone 85-968

Computer Line
Hiruharama House
TAUMARUNUI
Phone 8490

James Electronics Ltd
328 Pollen Street
THAMES
Phone 86-893

Cedar Save Furniture
Tristram Street
FRANKTON
Phone 82-997

Doug Morris Ltd
Great South Road
HUNTLY
Phone 86-113

Muir Radio
Arawa Street
MATAMATA
Phone 7468

Technology Resources Ltd
8 Thackeray Street
HAMILTON
Phone 393-801

Waikato Computer Centre
Princes Street
HAMILTON
Phone 393-416

BAY OF PLENTY
Compucation Systems
50 First Avenue
TAURANGA
Phone 88-433

Hamilton Sanyo
87 Eruera Street
ROTORUA
Phone 87-146

Jackson & Booth Ltd
Islington Street
KAWERAU
Phone 7030

HAWKES BAY
Progressive Computer Systems
Dalton Street
NAPIER
Phone 54-444

Timms Business Equipment Ltd
302 West Eastbourne Street
HASTINGS
Phone 69-906

Timms Business Equipment Ltd
Tennyson Street
NAPIER
Phone 54-250

GISBORNE
Triad Electrical Ltd
348 Palmerston Street
GISBORNE
Phone 5256

TARANAKI/WANGANUI
Laidlaw's Ltd
49 Liardet Street
NEW PLYMOUTH
Phone 83-667

Stewart Appliances
134A Victoria Avenue
WANGANUI
Phone 52-780

Taranaki Micro Electronics
Centre City Bldg, Devon St East
NEW PLYMOUTH
Phone 84-087

MANAWATU/WAIRARAPA
David Brice Electronics
47 Kimbolton Road
FEILDING
Phone 37-141

Colleges Computer Centre
170 Queen Street
MASTERTON
Phone 83-348

Lowrie & Watt Ltd
163–163 Rangitikei Street
PALMERSTON NORTH
Phone 84-934

Vaccari Electronics Ltd
306–308 Church Street
PALMERSTON NORTH
Phone 86-896

Wairarapa Camera Service
42 Queen Street
MASTERTON
Phone 84-657

WELLINGTON
Einstein Scientific
177 Willis Street
WELLINGTON
Phone 851-065

Ice Gear TV Services
187 Main Road
UPPER HUTT
Phone 282-475

Jenness Appl. & Music Centre
173 High Street
LOWER HUTT
Phone 661-145

Office Requisites Ltd
Cnr W Taylor & Featherston Sts
WELLINGTON
Phone 721-302

Public Service Investment Store
Waterloo Featherston Sts
WELLINGTON
Phone 725-123

NELSON/MARLBOROUGH
Hird Home Appliances
13 Charles Street
BLENHEIM
Phone 88-999

Personal Computer Systems
37 Toulouse Street
NELSON
Phone 79-302

Ross & Sang TV Services Ltd
58 Collingwood Street
NELSON
Phone 80-397

CANTERBURY
The Computer Centre
590a Colombo Street
CHRIST CHURCH
Phone 793-428

Crow Video Services Ltd
81 Ivory Street
RANGIORA
Phone 6230

Good Time Enterprises
200 Manchester Street
CHRIST CHURCH
Phone 797-611

SICOM Ltd
86 Oxford Terrace
CHRIST CHURCH
Phone 796-259

Smith & Church Electrical Ltd
Cass Street
ASHBURTON
Phone 89-019

David Wood Ltd
High Street Arcade
Hereford St
CHRIST CHURCH
Phone 797-226

Gill's Business Equipment
40 Stafford Street
TIMARU
Phone 84-211

WEST COAST

Graham Electronics
101 Revell St
Hokitika
Phone 384

OTAGO/SOUTHLAND
Andersons Electrical Ltd
29 Helwick St, Wanaka
CENTRAL OTAGO
Phone 7623

Eclipse Radio & Hobbies
134–136 Stuart Street
DUNEDIN
Phone 778-102

Eastern Southland Computers
C/- Robin Isaac, Medway St
GORE
Phone 5717

Ken France Electronics
46 Tarbert St
Alexandra
Phone 5021

OES Business Systems Ltd
135 Dee Street
INVERCARGILL
Phone 84-466

Otago Business Equipment Ltd
Cnr Liverpool & Bond Streets
DUNEDIN
Phone 776-227

Stand Vic House
24 Bremner Road, Fairfield
DUNEDIN
Phone 881-439

inside BITS & BYTES . .

Part two of Nick Smythe's article on school microcomputer networks has been deferred until the April issue due to the release of the Education Department report on computers in schools.

EDITORIAL

Heads in the silica

After a long and secret labour the elephant has delivered itself of a mouse. The long-awaited (nay long despaired) recommendations for computer support in schools is out.

These recommendations have not been lightly entered in to. The list of consultants and civil servants involved in the whole decision is long. The bill (in terms of wages, Poly development and so on) is not inconsequential. The result, in terms of value for money, is laughable. A sixth-former with a set of computing magazines might have done the same in a free period, although he would be more likely to do so while standing in line for a turn on a much-overused keyboard.

The final outcome of the deliberations of the Minister of Education (and friends) is to support (in the loosest possible sense) five computers as school machines. The Apple II, the Poly, the BBC microcomputer, the NEC PC 8000 and the BMC 800. These are simply recommendations. Without financial support or other commitment were these recommendations worth while?

What might one have expected from a search for a suitable school-computer policy? Probably the first priority might have been to co-ordinate school use so that teaching programs could be interchanged. Well, five machines are better than 25 but look to far larger educational systems than ours. In general they recommend (and fund and support) two or three systems. England and Wales support three locally produced systems and most Australian states support one or two machines officially. What price five machines in a small nation?

Then you might look for building on an existing base of school machines. There are arguments against following old precedents but one can not help but feel sorry for the schools with TRS-80/System 80 type machines or Commodore equipment. Of the five recommended machines three have yet to make any significant presence in schools. The BBC machine may well ride in on the tails of the forthcoming television series but it would be good to hear from schools with existing NEC and BMC equipement and learn how they cope with getting software.

Another priority might be to use standardisation as a means of getting cheap equipment for schools

Turn to page 18

Coming up in BITS & BYTES

Micros in Architecture

Micros in court — A look at solicitors' packages.

Hardware reviews
- **IBM PC**
- **NEC PC-8000**

Computer camps — Hundreds of Auckland school children spent a week of their holidays learning about computers.

Plus columns on:
- **Farming**
- **Education**
- **Business**
- **Beginners**

BITS & BYTES is published monthly, except January, by Neill Birss, Dion Crooks and Paul Crooks. Editorial and subscriber inquiries to Post Office Box 827, Christchurch.

ADVERTISING:

Auckland — Wendy Whitehead. Phone 794-807
Wellington — Annie Carrad: Phone 723-431
South Island — Graham Beecroft: Phone 554-265 Chch or telephone Christchurch 66-566

EDITORIAL:
Auckland — Cathy and Selwyn Arrow: 30A Bracken Ave, Takapuna
Christchurch — P.O. Box 827 or Phone 66-566
WELLINGTON — Shayne Doyle, 280-333, ext 892(W), 278-545(H).

Production: Roger Browning, Graeme Patterson, Lee Teck Fui, Janine Morrell
Typesetting: Gail Rankin
Cover and Graphics: Sally Williams

Technical Editor: Chris O'Donoghue

Subscription rate: $8 a year (11 issues) adults
$6 a year for school pupils

Subscriptions begin from the next issue of BITS & BYTES after the subscription is received.

Back copies are available (except issue 1) at $1 per copy plus 50c per copy postage and packaging.

Subscription addresses:
When sending in subscriptions please include postal zones for the cities. If your label is incorrectly addressed please send it to us with the correction marked.

Distribution inquiries: Bookshops — Gordon and Gotch, Ltd. Computer stores — direct to the publishers.

Printed in Christchurch by D.N. Adams, Ltd.

MICRO NEWS

Solid State Equipment Ltd, of Lower Hutt are manufacturing a range of PACIFIC field data loggers designed for all harsh environments. Hermetically sealed, with a tough fibreglass case the logger is powered from a 12 volt battery using about 10 Ah per month.

It accepts information from electrical inputs and stores it on a computer grade cassette. This cassette can either be read by a separate reader or by the logger itself, producing serial ASCII data for further analysis or printing.

The process of data logging can be considered as recording information from a number of channels. Each channel has associated with it a set of parameters.

The unit is designed as a general purpose logger with a wide range of applicability. Operation is simple, it is easy to program using a keypad and LCD and the programs can be saved on cassette for future use.

The instrument is designed and built locally, has had many machine years of use, and can be built with special routines to handle special needs. It can be used industrially, in a laboratory or in the field.

* * * *

LISA, Apple's new personal computer, is expected to have special appeal for senior businessmen and decision-makers because of its simplified method of operation.

Instead of having to adopt rigid conventions and special languages, the user works naturally with familiar objects and tasks that are depicted as pictures on LISA's screen.

LISA utilizes a "mouse", a palm-sized device that positions the cursor, or pointer, anywhere on the screen. By moving the mouse on the desktop, the user points to and selects objects on the screen.

The LISA screen features graphic representations of objects typically found in an office environment — stationery, a clipboard, a wastebasket, and more. These pictures also represent tasks — writing, setting aside documents on the desktop, putting documents back in folders, or throwing something away. Collectively, these objects form the "desktop manager", the part* of LISA's software architecture that organizes information, documents, and tasks just as the user would in a typical office.

LISA will retail in New Zealand for about $20,000.

* * * *

Apple have released an upgraded version of the Apple II Computer — the Apple IIe.

The Apple IIe is expected to retail for about the same price as the present Apple II.

"The new Apple IIe includes 64 K of random access memory (expandable to 128K with a further 64K), upper and lower case characters, and an expanded keyboard," said Mr Eardley-Wilmot, managing director of CED Distributors, New Zealand Apple distributors.

"The system offers improved logic board and case design, and will accept a low-cost 80-column card for text editing.

"The newly designed main logic board of the IIe uses one quarter as many integrated circuits as the previous Apple II model. Two new LSI (large scale integrated) circuits, custom-designed for the IIe by Apple engineers, replace approximately 80 separate integrated circuits used in earlier models," said Mr Eardley-Wilmot.

Apple is introducing new software programmes that have been designed to use all IIe enhancements, including the additional memory. These include the Apple Writer II word processing programme, and Quick File II, information management programme."

* * * *

"COMPU-PAD", a pad of forms designed to ease the programmers burden, has been released in New Zealand. Marketed by a Dunedin-based company, Leading Edge Computers Ltd, the pad consists of coding sheets, printer and V.D.U. layouts.

The coding sheets whilst designed primarily for BASIC can also be used for most other languages. They encourage the program to be written on paper and thought through before it disappears into the depths of computer memory. Read-ability techniques such as indenting loops are easily implemented on the squared layout.

The V.D.U. layouts cater for 80x24 and 16x64 screen sizes as well as chunky style graphics. Smaller screen sizes are catered for by using only a section of the form.

The printer layouts are drawn to 10 Characters per Inch to allow actual printouts to be easily measured against planned output during testing.

* * * *

The new Commodore 64 computer is expected to be on sale in New Zealand before the middle of the year.

Released in the United States towards the end of last year, the Commodore 64 is reported to be selling well at $595 per unit.

Here the price will be around $1600 but (as the name implies) that includes 64K RAM as standard plus all interfaces built in (serial, parallel and disc drives). The Commodore 64 runs all VIC peripherals as standard.

Of interest to music buffs will be the Sound Interface Device Chip which Commodore claims

Turn to page 30

Offerings from the giants

IBM . . . .

In case you hadn't noticed the Charlie Chaplin look-a-like adds, IBM have released their personal computer in New Zealand.

The release here comes 18 months after the United States launch — the delay being caused by an inability to meet demand in the US say IBM.

According to reports the IBM PC is now the largest selling personal computer in the US (outselling Apple and Radio Shack offerings) in spite of earlier claims that it was overpriced.

Similar claims are likely to be made in New Zealand with the basic unit containing 64K RAM and one 320K disk drive costing $5107.

But priced separately are the keyboard ($587), monochrome display screen ($732), adapter card for the screen and a printer ($699) and the power cable ($35). This gives a total of $7160 before software or a printer is added.

At this price, the PC, as IBM admit, is aimed at the small business market.

IBM see a typical configuration in a New Zealand business as:
Hardware — 128K RAM, two 320K diskettes, keyboard, screen, and an IBM 80 characters per second dot matrix printer. Cost $10,932.
Software — IBM Disk Operating System, word processing and four accounting modules. Cost $4471.
Total cost $15,403.

Based on a 16 bit Intel 8088 microprocessor, the IBM PC has 40K ROM, 64K RAM (expandable internally to 544K), an 83 key keyboard (including numeric keypad and 10 function keys) and 25 line by 80 character wide screen.

Five expansion slots are provided for additional memory, monochrome screen, colour/graphics display, printer, communications, games adaptor and so on. All of these require separate cards (each separately priced) as none of the slots are dedicated. The advantage of this says IBM is that users can expand to their own requirements with no constraints.

Available from the US is a card called Baby Blue which allows the IBM PC to run CP/M 80 (8 bit) software.

No hard disc is offered at this stage.

On the software side the PC comes with standard Microsoft BASIC in ROM and a choice of three disk operating systems (again each priced separately), IBM DOS, CP/M 86 and UCSD p-system.

A number of software packages are offered including financial spread-sheets, word processor, educational and games and a suite of business programs from Interactive Applications of Auckland.

To assist the development of New Zealand software for the PC, IBM have established a software publishing department which will review software submissions from authors and if acceptable market them.

A lot of emphasis will be placed on user-friendly software say IBM.

Carry-in service for the PC will be provided by dealers throughout the country and IBM product centres in Auckland, Wellington and Christchurch. There will be no on-site service.

ICL . . . .

ICL has introduced a 1983 model of its personal computer — or to be more correct four new models, the 15, 25, 26, and 35.

The enhancements on the first model released six months ago include:

- 70 per cent more processing power with the 5MHz 8085A microprocessor.
- 300 per cent more diskette capacity — 798k formatted.
- 100 per cent more hard disc capacity on the model 35 — 100 megabytes.
- More internal memory — up to 512K RAM on all models.
- More expansion — four RS-232 ports with the option of 8 on all models.

Later this year ICL will have:

- Colour monitor.
- Upgradeable to 16 bit (8088 processor) operation.
- ICL and IBM communications.
- Additional plug-in hard discs.

Two other interesting features of the new ICL models are disc performance improvements called cache memory and virtual disc.

Cache memory is an optional large disc buffer allocated in RAM. This is used to minimise the number of read/write accesses to a disc. Whenever a disc access is requested the cache memory is first searched for the information. If the information is in cache the program will be serviced faster say ICL.

Virtual disc is a simulated drive in RAM. It appears to the user program as a conventional disc but because the files are actually held in a special RAM, access is much faster say ICL.
Virtual disc is available in 256 or 512K formatted versions. This is over and above the normal directly addressable RAM.

Prices for the new ICL models:
Model 15, (64K RAM, two diskettes), $5500.
Model 25, (64K RAM, one diskette, one 5Mb hard disk), $10,700.
Model 26, (256K RAM, one diskette, 5Mb hard disk), $11,800.
Model 35 (256K RAM, one diskette, 10Mb hard disk), $13,200.

Our first winner

WAYNE DOBSON

Wayne Dobson, a sixth former at Karamu High School, Hastings, is the winner of "BITS & BYTES" competition No.1. He wins a Sinclair ZX81 provided by David Reid Electronics, Ltd.

Wayne's entry was a Sinclair program which will be printed in "BITS & BYTES" next month.

'Equation' is for a ZX81 with 16K RAM, and unexpanded ROM.

Karamu High lets pupils take Sinclairs home for practice involving the whole family (see article by N.R. Moir in "BITS & BYTES", No. 2, October).

Wayne, who was 15 when he entered the contest, has been working with computers for two years at Karamu, but until now has not had a machine of his own.

"Mostly I like writing games and learning about languages; my favourite languages are BASIC and Forth," he says. His favourite ZX81 game is 3D Monster Maze, by Melbourne House Software, Inc.

This year Wayne hopes to pass University Entrance, so that he can study programming full time at polytechnic next year.

Wayne was the unanimous choice of our judges for first prize from the 22 accepted entries for the competition, which was for an original game by a school pupil. Besides the David Reid Electronics prize, two fine books have been given by the Australian and New Zealand Book Company for runner-up prizes.

The books are "From Chips to Systems: An Introduction to Microprocessors," by Rodney Zaks, published by Sybex, and each with a retail value of $31.95.

The runners-up are: G.P. McKenzie, of Snodgrass Road, R.D.2, Tauranga ("Haunted Mansion", for a TRS-80 with 16K; and Kevin Berry, of 48 Hope Street, Christchurch, 1 (a "Horserace" for an Apple).

"BITS & BYTES" thanks David Reid Electronics and the A.N.Z. Book Company for the prizes, and thanks all competitors who entered.

Entrants' listings, tapes, etc, will be returned as soon as possible.

Turn page sideways for a revelation in programming...

PROFESSIONS

Laboratory Computing

— a chemist's view of micros

By CATHY ARROW

"Speed of operation is the most important factor," says Gavin Murdoch of Hall Avenue Chemists, Otahuhu.

He commenced using an Apple Computer in his pharmacy business three years ago. At that time suitable programs for chemists were very rare and relatively expensive. He therefore opted to write his own. Since then he has extensively revised his original program for optimum efficiency.

From the time the patient hands over a prescription until the label is processed, takes one to two minutes depending on the number of different crugs or mixtures prescribed. Repeats only take a few seconds.

Gavin currently utilizes an Apple II Computer, with 8" floppy disk and a Centronics 779 printer for his entire business operation.

His initial setting up for the dispensary required the typing in of information, that included over 2,000 drugs. The data is now available alphabetically on two files — one by trade names and the other by generic names.

His series of programmes use approx 40k of memory and provide for a wide range of business activities that include: accounts, payment, cashflow, bankings, creditors ledger, cash analysis, tax, wages, credit/purchase calculations, analysis of purchases and shop retail margins. Plus a full range of specialised pharmaceutical activities such as: alphabetic drug holdings file, prescription dispensing, re-order list for drugs, and update EFP (extra fee on prescription). These are charges notified monthly by the Chemists Guild and printed on the label where applicable.

Each label details:

Line 1. Quantity and type of medicine.
Line 2. Dosage.
Line 3. Name and address of patient.
Line 4. TN (transaction no.), DN (doctor no.), PP (prescription no.), PD (prescription date), DD (dispensing date).
Line 5. Repeat details.

All information is saved to disk as work is done, to avoid loss if there is a power flick, which does occasionally happen.

Usually twice a day dispensary activities are printed out and filed as a paper record. This includes: wholesale cost, selling price (for Social Security), EFP charge, NSS (not on Social Security), dispensing fee, container cost, no. of prescriptions dispensed.

Gavin has 15 relatives who are, or have been, chemists. He has become an avid programmer and is happy to receive commercial enquiries.

In his spare time he has also written a comprehensive hotel/motel management program covering reservations, finance and restaurant operations. Another of Gavin's programming efforts is writing a bridge learning program

His comprehensive pharmacy programmes are written in BASIC as well as machine language. They certainly serve all his needs and the personal pleasure he gains in using his computer located in the dispensary is obvious.

DICK SMITH

NEW ZEALAND

SAVE $300

NEW ZEALAND'S MOST POPULAR PERSONAL COMPUTER

THE DICK SMITH SYSTEM 80

It hasn't become Australasia's fastest selling home computer by accident! The System 80 is a complete, ready-to-go computer. You don't have to buy add-ons like cassette recorders or monitors: the System 80 comes with a built-in recorder and modulator, so you can use it with any TV!

Don't be fooled by inferior computers which don't offer you the incredible features of the System 80 (like software compatibility with the world's largest range!) The System 80 is not just affordable: it's usable as well !!!

Compare these features and price with any other computer.

- Flashing Cursor.
- Built-in speaker and amplifier for programs with sound effects.
- Full upper & lower case video display capability.
- Monitor program (suitable for the hobbyist to program in machine language)
- Screen print facility (obtain a printout of any 'page' appearing on the screen)
- Three month guarantee from date of purchase.
- Great expansion capability and flexibility
- No separate tape recorder to buy – it's in-built!
- Cassette level meter and provision for a second cassette.

Cat. X-4005

Was ~~$1295~~ now only

$995

OVER 10,000 SYSTEM 80's SOLD

Take advantage of this once-in-a-lifetime special offer . . . Bulk purchase has allowed us to sash the price on this popular computer!!!

"The best value around for a ready built computer . . ."
(Electronics Aust. Magazine)

BIG SAVINGS ALSO AVAILABLE ON THE 'BUSINESS' VERSION SYSTEM 80 WAS ~~$1245~~ NOW $995

Cat. X-4100

New Zealand's least expensive Business Computer! Use it for Word Processing, Accounting, Filing, Invoicing, Stock Control, etc. It has a separate 15-key numeric keypad, plus 4 special function keys. Why pay thousands of dollars more when the Business Computer System 80 MKII does it all? Cat. X-4100

7 DAY TRIAL OFFER
If you're not completely happy, you can return the System 80 in original condition & packing for a full refund!

DSE/EA453/JW

DICK SMITH Electronics

98 Carlton Gore Rd.
Newmarket,
Auckland 1.
Ph: 504 409

MAIL ORDERS:
P&P $5.00

Mail Orders, Private Bag, NEWMARKET

Shop by phone.
Use your Bankcard.
Or Visa card.

The logic of cassette tapes

By E. M. FREEMAN

If you belong to the majority of New Zealand home computer enthusiasts forced to plod into the world of micros with an audio cassette storage system, this article is for you.

Most of us have been frustrated with the almost meaningless error messages displayed when trying to read the last copy of our latest development projects. To understand them it is necessary to know something of the logic and circuitry involved in reading/writing cassette tapes.

Cassette systems come in two shapes: audio systems noted for their slowness and low price; and more sophisticated digital types, which generally allow full program control, directory functions, and higher baud rates.

Most audio cassettes use a variation of two formats for representing logical data, of which frequency shift keying or Kansas City formatting is the most common. The Kansas City format (called after the convention where it was defined) uses 8 cycles of 2400Hz signal to indicate a logical "1" and 4 cycles of 1200Hz to indicate a logical "0". The complete sequence for a single character consists of one start bit, eight data bits, and one or more stop bits. If one stop bit is used we get a total of 10 bits for a maximum baud rate of 300 or 30 characters per second. For those of you who have forgotten the term baud (named after Emile Baud, inventor of the five bit Baudot code) is a measure of the total signalling rate, that is all start/stop bits included, whereas the programmer normally rates a device in bytes or characters/sec. Various manufacturers have implemented this format in rates up to 1200 baud by reducing the cycle count in each bit. Atari uses 1200 baud for the 400/800 while Commodore's VIC 20 uses 300 baud.

The less common Tarbell format is an example of frequency/double frequency recording. A logical "1" is represented by two signal transitions and logical "0" by no signal transitions in each "bit cell".

Now all we need is a device to interface between the digital CPU and the analog cassette. These have also been standardised and range from the UART or USART (universal serial asynchronous receiver/transmitter), SCC (serial communications controller) to the more general SIO (serial input/output) devices. (The term, asynchronous, refers to the lack of a clocking signal in the data-transfer operations in contrast to communications protocols that require more precise synchronisation between sender and receiver).

Now we have something to blame for all those innocuous errors in reading tapes for it is generally in the SIO where parity, checksum, framing, and overrun conditions are detected.

Parity refers to the count of logical "1"s in the transmitted character. It is generally implemented using the eighth bit in the seven-bit ASCII code or as a hardware function in eight-bit codes used on mainframe computers. Two forms are commonly defined: Even parity where the eighth bit is turned off or on to make the count of "1" bits even, and odd parity where the eighth bit is turned on or off to make the count of "1" bits odd.

Checksum errors can be implemented in a number of ways but a useful technique involves "exclusive or-ing" each successive byte with its predecessor and

retaining the result. Depending on the system designer this may only detect an error at the end of the run or at successive intervals. This type of checksum is usually a function of the CPU tape-loading program rather than the SIO.

Overrun results from the device detecting a character before the previous character has been read by the processor. This results in the loss of at least one and usually more incoming characters.

The group of start/stop and data bits is referred to as frame which leads to the last common error of the same name. Framing errors result from the loss of stop bits in the transmitted frame and are generally an indication of mismatched operating frequencies between transmitter and receiver. (To you and me it results from writing data without the record button depressed.)

It is apparent that the loss of a single bit may cause any of these undesirable conditions. As a single 100-line BASIC program may contain 50,000 to 100,000 bits, this can be very frustrating indeed.

Unfortunately, most designers err on the side of caution and halt the read when any error is detected without trying to salvage the following data, allowing the application programmer to correct what may be a single misread character. (All those crying 'No': This is reasonably safe in the home environment with BASIC being predominant). Combine this with an operating system which clears RAM before each read and you had best start looking for solutions.

Some small systems, notably Commodore's VIC 20, Sinclair's ZX series and Texas Instruments' 99/4A, allow verification of the written data but for those less fortunate multiple copies on high quality tapes result in fewer reprogramming efforts.

How to prevent the errors in the first place?

Keep the readwrite heads clean. Ask any professional who has used cassette tape systems (rare as they are) and you'll find the heads are cleaned daily. You should clean them once a month using the age old method of cotton swabs and alcohol (get the common 95 per cent denatured variety) to wipe all heads, capstan, and pinch roller. Cheap head cleaners are available which do a similar job with less effort but don't use them for too long at one cleaning as they are somewhat more abrasive than ordinary tape.

Next, look at the tapes you use. Do they cost a dollar and make an ungodly noise when used? Chances are they may have thin oxide coatings or transport mechanisms that do not hold the tape parallel against the heads. Generally speaking, medium-priced, name-brand audio tapes using ferric oxide (not chromium oxide) as a recording medium are more than adequate.

Tapes stretch and contract in response to temperature. An unreadable tape may become readable when re-read at the temperature it was recorded in. More often than not tapes left in direct sunlight will be unreadable. Also avoid recording in high humidity.

Tapes are a magnetic recording medium so don't leave them on or near televisions, telephones, radios, etc.

If all else fails when did you last drop your recorder? The azimuth or 'angular setting' of the heads may need resetting as even a slight malalignment may result in 'fuzzy' recordings.

For those of you who like punishment, it's a simple matter to put indexes at the front of longer tapes, but the time required to read them will soon put even the most patient off the idea. It is far better to use paper indexes and C10 tapes (recording on both sides) with as many copies per side as will fit. For development work longer tapes up to C45 or C60 can be used to intermittently back up your work as you go.

Thereby when the inevitable lockup occurs (take note, Atari owners) all is not lost. One reprogramming effort will convince you of the worth of this regime (grab a cup of coffee on each back-up).

One last hint: fast forward and rewind new tapes to loosen the windings and remove stray static charge.

Start saving now for a disk drive!

Beat loading troubles

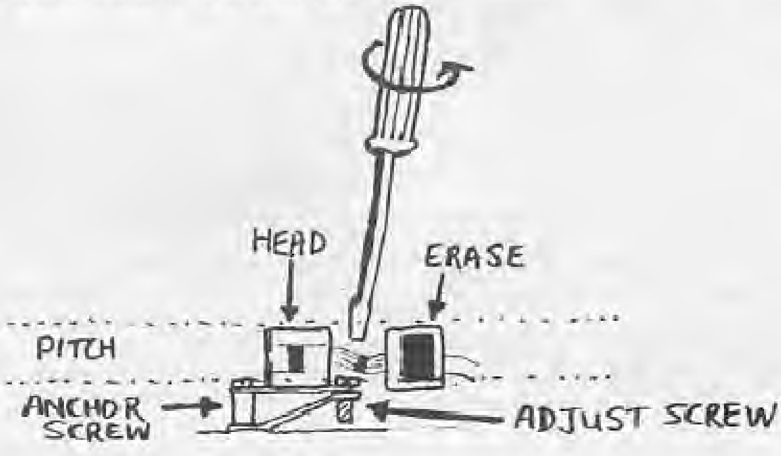
This is a tip from Paul Kinley for those having trouble loading programs from cassette.

If you have trouble loading programs, more often than not the tape head is mis-aligned. Using a small screw driver adjust the pitch screw.

Recorder heads appear to be set at slightly different pitches. Audibly it doesn't sound too bad, but for digital processors it is lousy.

To solve the problem, run the bad tape volume up, then adjust the tape head pitch screw until the sound is sharp and clear. It should now load; you may have to re-adjust to play your old tapes.

This will work if you can play a tape with the lid up. I actually drilled a hole in the case to get access to the screw.



Banish the cassette tape blues

By STEVEN DARNOLD

Most of the low-priced computers in New Zealand use cassette tapes for storing programs and data. Certainly, disk drives are available, but often a disk drive costs more than the computer itself. The Commodore VIC, for example, sells for $795 while its disk drive costs $1295. A schoolboy may be able to scrape up enough money to buy a VIC, but a disk drive is just a far-off dream.

Some computers (Atari, Commodore, System 80) use specially designed cassette drives, others use ordinary cassette recorders. The specially designed drives are usually more reliable and smoother to operate, but they are still subject to the limitations of cassette tapes.

There are three main problems in using cassette tapes with computers. First, because the tape moves slowly, the rate of data transfer is low. Second, because data on a cassette is stored in one long strip, there may be delays in finding a particular item. Third, because of inadequate care, a cassette tape may lose the data stored on it.

The first problem is unavoidable. The tape passes over the heads at a standard speed of 5 centimetres per second, and less than 100 bytes can be stored per centimetre. Fast tape systems (e.g. the "Stringy Floppy") are available, but they are considerably more expensive than standard cassette drives.

The second problem occurs when several programs or data files are stored on a single tape. A lot of time can be wasted hunting for a particular entry (just rewinding the tape can take two minutes!). This problem is easily eliminated by using short tapes and recording only one program or file per side. As a result, there is no searching, and a rewind takes less than 20 seconds. Of course, if you put only two programs on each tape, you will need a lot of tapes.

Make your own

Here is a recipe for making inexpensive C-10 tapes:

1. Buy a good-quality C-60 tape (cost about $4). Make sure it is sealed with screws, not glue.
2. Buy five empty cassette housings. They should cost about $1 each.
3. Play the C-60 for five minutes. Then remove the screws.
4. Open the C-60 and cut the tape.
5. Take the 5-minute spool of tape from the C 60 and put it in one of the housings. Connect it to one of the two empty spools in the housing.
6. Take the other empty spool from the housing and connect it to the full spool in the C-60.
7. Press the C-60 together (it is not necessary to screw it shut) and play it for five minutes.
8. Repeat steps 4 to 7 until there is 5 minutes of tape in each of the housings and 5 minutes left in the C-60.

Ten tips

The third problem with cassette tapes is unreliability. Load errors occur, and users tend to blame the system. However, most reliability problems arise from careless use and can readily be eliminated.

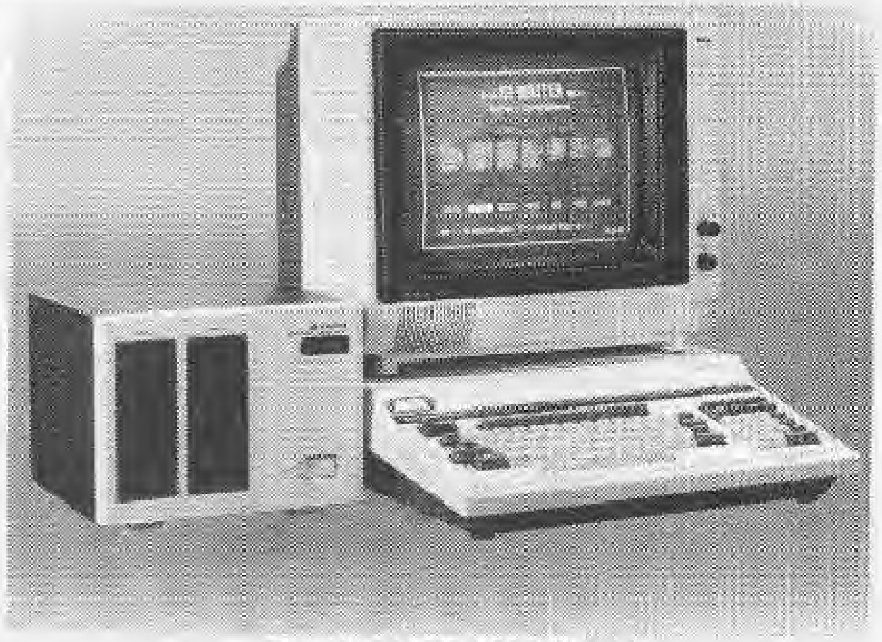
The following suggestions will improve the operation of any cassette system.

1. Use good quality tapes. Data will be lost if there are splices, scratches, or thin spots. TEAC SDR tapes are very reliable and reasonably priced.
2. The length of the tape should be C-60 or shorter. Longer tapes may be thinner and consequently less reliable.
3. Brand new tapes need breaking in. Before using a tape for the first time, fast-forward it to the end and then rewind it.
4. Use only one side of the tape or, alternatively, use only the first half of each side. Otherwise, data on one side may be affected by data recorded on the other.
5. Avoid stopping the computer in the middle of a save. This may magnetize the cassette heads.
6. Always rewind tapes immediately. That way, any scratching or stretching of the tape will fall on the leader, not on the data.
7. Return tapes to their boxes and store them well away from transformers, television sets, telephones, and other electrical appliances. Magnetic fields can destroy data.
8. Clean the cassette heads and pinch-roller every week. A cotton bud with isopropyl alcohol (from your chemist) works fine. If the bud is discoloured after cleaning the heads, then either you aren't cleaning often enough, or one of your tapes is flaking badly (get rid of it).
9. Demagnetize the cassette heads regularly. Otherwise, a magnetic charge builds up and partially erases tapes passing over the heads. Eventually the signals on these tapes will be too weak for the computer to pick up. Demagnetizers cost about $30 and can be purchased from audio equipment shops. If $30 is too steep, find a friend with a demagnetizer and visit him once a month with your cassette recorder.
10. From time to time, re-record the data on the tape. There is always some deterioration over time, so it pays to refresh the tapes before they fade. If your system has a verify command or an error log, use it. It's a good way to tell when a tape needs re-recording (or discarding).

If you follow the above 10 points and still have problems, take the cassette recorder to your dealer. The heads may be mis-aligned or there may be a mechanical fault.

Banish the cassette tape blues. Organise your system along the lines recommended here, and you will find that cassette tapes can be both convenient and reliable.

The Hitachi MB-6890 (Peach)



The Hitachi MB-6890 (Peach)

By WARREN MARETT

Is the Hitachi MB-6890 a "peach" of a machine? By name it is, but in terms of features it is a middle of the road personal and small-business microcomputer.

According to Colin Bligh, at AWA Data Systems, the MB-6890 was given the name "Peach" in Australia by marketing people hoping to jump on the Apple band waggon. In fact, the machine is not compatible with the Apple and was not intended to be.

Released a few months ago in New Zealand, the MB-6890 is primarily aimed at the small-business user. Later this year AWA will be releasing another Hitachi microcomputer, this one will be a 16-bit machine compatible with the IBM PC.

The MB-6890 is an 8-bit machine which uses a 6809 microprocessor made by Hitachi under licence to Motorola. Hitachi, a major Japanese mainframe manufacturer, has only recently moved into micros.

Released in Australia a year ago as a test market, the MB-6890 has been picked up by many Australian dealers and has been receiving plenty of publicity. Much of the software and documentation sold by AWA in New Zealand has been developed or adapted in Australia.

In New Zealand the MB-6890 is being distributed by AWA Data Systems through dealers. The cheapest configuration (for use with a cassette recorder), at about $2600, is the CPU keyboard unit and a green monitor.

A basic configuration for business use, with a colour screen and dual 5¼ inch floppies and an 80 cps printer will cost just over $8000. These floppies hold 640K bytes of data in total. The 8 inch floppies, costing an additional $2570, hold a handy 2 megabytes of data in total.

So the MB-6890 is competitively priced, but does not break any price barriers. Its value largely lies in solid construction, a good keyboard, a useful colour

graphics facility, a good range of rear connectors (colour monitor, black and white video, parallel printer, audio cassette, light pen and serial interface, with six expansion slots for other interfaces) and plenty of memory as standard (24K ROM, 32K RAM).

The keyboard has a numeric key pad and five function keys, which with the use of the shift key, give ten programmable function keys. The function keys hold standard character strings (such as "RUN") at boot-up but can be loaded with other strings through a user program.

Your television set cannot be hooked to the MB-6890 through a standard connector — the screen must be an RGB colour monitor or a monochrome video monitor. The green phosphor monitor available for $510 from AWA has poor contrast and could not be recommended.

No complaints about the colour screen. It is easy to read and the colour graphics demonstrations look impressive. A maximum of 640 x 200 dot resolution is selectable (which takes 16K bytes of RAM away from you) but lesser resolutions are selectable with consequent lower memory overheads.

Your reviewer is pretty ignorant when it comes to colour graphics, so he can't offer too many comments about the MB-6890's graphics features. Apparently there are a good range of commands for controlling the graphics. Colour is chosen at a character position level by providing a 5-bit code which is loaded into a separate 5-bit x 16K colour memory.

Text characters are written to the screen at 40 or 80 characters per line with 25 lines per screen.

Microsoft level 3 BASIC is provided in the 24K byte ROM in the MB-6890. A number of extensions have been added to support the hardware capabilities of the MB-6890 and a screen editor is included. The disc monitor system adds additional BASIC features.

Colin Bligh frankly admitted that the BASIC is not very fast — fancy screen work will require you to resort to machine language.

The standard of documentation is disappointing and quite off-putting. The BASIC manual and Disc BASIC manuals supplied by Hitachi are poorly presented and written in a confusing Japanese version of English. In the first few paragraphs read there were unresolved section and diagram references and there are no indexes to help find your way around. The manuals are not detailed enough in some areas, such as graphics programming and basic introductions to the machine and its software.

AWA does have for sale an Australian-produced alternative to the BASIC manual. This should be purchased. Another Australian source has produced a guide through the MB-6890 ROM. Hitachi optionally provides a technical manual.

AWA can provide a word processing package, a spreadsheet modelling program, a series of financial accounting modules, a data management package, an assembler, a disassembler and about 25 assorted games and demonstration programs.

Much of the marketing emphasis will be on the Australian-produced financial modules, including debtors, creditors, invoicing, general ledger and inventory.

A CP/M card will be offered but compatibility with CP/M is not an option that will be emphasised.

The word processing package, adapted in Australia from an American package, looks a superior package for a machine in this range. It is able to use colour on the colour monitor — a good way to distinguish control characters from text.

AWA will be offering service contracts for the machine directly with the end-user customer.

Review unit from:

AWA (New Zealand) Limited,
P.O. Box 50-248, Porirua.

Microcomputer Summary

Name:	MB-6890 Personal Computer
Manufacturer:	Hitachi Corporation, Japan.
Microprocessor:	HD6809
Clock Speed:	4MHz.
RAM:	32K standard, 30K or less usable.
ROM:	24K.
Input/Output:	Audio cassette interface, 600 baud; Printer interface (Centronics, parallel); Light pen interface; RS232 serial interface, 300—4800 baud; RGB colour monitor interface; Black and white video signal connector.
Keyboard:	87 key, QWERTY standard typewriter-style keyboard.
Display:	40 characters by 25 lines or 80 characters by 25.
Languages:	Microsoft level 3 BASIC with extensions.
Graphics:	Up to 640 X 200 dots, seven colours.
Sound:	Speaker with volume control.
Cost:	$2150 for CPU and keyboard unit (not incl. manuals).
Options:	Six edge connectors for I/O expansion; Two extra memory sockets, each 16K, each $224.
Peripherals:	Colour monitor, $1687 Green monitor, $510 Light pen, $520 8 inch dual DSDD floppy drives and controller, 2M bytes total, $6086; 5¼ inch dual DSDD floppy drives and controller, 640K bytes total, $3515; Star 80 cps printer, $1199.

Small business computing

By JOHN J. VARGO

Here is a summary of the application areas discussed last month for easy comparison.

APPLICATIONS	Shorten Billing Cycle	Lower Inventory	Increase Sales	Control Costs	Manage Cash	Reduce Labour Expense	Plan and Control Growth
Inventory Control	x	x	x	x	x	x	x
Creditors				x	x	x	x
Debtors	x	x	x		x	x	x
General Ledger			x	x		x	x
Word Processing			x	x		x	x
Business Modelling			x	x	x	x	x

Table adapted from Data General's Insider's Guide to Small Business Computers. 1980.

It is necessary to quantify these prospective benefits in dollar terms for your particular business. Here is a hypothetical example for illustration.

C.J.B. Ltd is a materials supplier to the construction industry. After a thorough evaluation of their information system they have concluded that they would benefit from a small business computer.

In particular, the following applications look promising:

- Inventory control/order entry
- Debtors control

C.J.B. management have discovered that they lose 3% of their total sales annually due to low and out of stock conditions. Additionally they believe they could reduce inventory levels overall by 5% through improved ordering and reduction of slow moving stock.

The expected financial benefits are computed as follows:

Increased profit =
Annual sales x .03 x Gross Profit Margin
= 850,000 x .03 x .15 = $3825

Interest savings =
Inventory total x .05 x Annual interest rate on loans
= 240,000 x .05 x .02 = $2400

In the area of Debtors control management believes that better control over credit approval and collection would give the following benefits:

Reduced bad debts =
Annual credit sales x 1/3 of current loss per cent
= 850,000 x 1/3 x .01 = $2830

Interest savings =
Average Debtors total x 8% reduction in collection period x annual interest rate on loans
= 200,000 x .08 x .20 = 3200

With all financial benefits added together the total projected annual benefit is $12,255.

EVALUATING THE PROPOSALS

Once the proposals have come in from the various vendors it is time to evaluate them. This should be done in a systematic fashion.

Here is a suggested series of steps to follow:

1. Establish measurement criteria and weightings (should be closely tied to the RFP) together with minimum qualification.
2. Determine your maximum price based on your earlier benefits analysis (cost savings, improved cash flow etc), and minimum performance points. This will eliminate the very expensive systems, and those that will not meet your minimum requirements.
3. Evaluate all proposals that meet the minimum qualifications in an evaluation matrix.
4. Choose your short list based on a cost/effectiveness ratio (top 3–5 scores), and contact current users of the systems to determine reliability and level of service.
5. Run your own test data on the short list systems and see if the system can handle your type of transactions at a speed satisfactory to you and produce reports of the right type, style etc.
6. Write up a contract, including all the elements that you expect to be contained in the cost of the system.

Let us take a closer look at these steps. Below is an illustration of an evaluation matrix[2] for a small business that does not expect to hire a programmer, and therefore

expects a "turnkey" system (one that is complete and ready to "turn-the-key" and start running).

Included in the matrix are steps one through four. A total of 1000 evaluation points was chosen arbitrarily to indicate the ideal system meeting all specifications.

This Matrix was adapted in part from a forthcoming monograph for the New Zealand Computer Society, entitled CHOOSING YOUR FIRST COMPUTER SYSTEM, by K. Ian Mitchell and Bevan J. Clarke.

EVALUATION MATRIX

System Components	Max. Eval. Pts.	Ideal System	MBI 371	Syst. 5800	Mylos 9000	Moly 80A	Hewlard 86
HARDWARE CONFIGURATION							
No. of workstations	75	5	4	5	6	5	4
Memory size Kb	75	512	256	512	720	120	512
Disc storage Mb	75	40	20	30	45	10	60
Keyboard Quality[1]	25	10	4	6	5	6	9
Video Quality[1]	25	10	6	6	7	10	9
Hi Res graphics[1]	50	10	0	8	0	10	9
Printers — No.	25	3	3	3	3	3	3
— Speed (CPS)	25	150	180	150	120	150	180
— Quality	25	Letter	Letter	Letter	Dot Matrix	Letter	Letter
Total	400	400	235	351	300	276	375
APPLICATION SOFTWARE[2]							
Inventory control	100	10	8	8	5	4	9
General Ledger	75	10	8	8	5	3	10
A/R Billing	75	10	8	8	5	4	9
A/P Cash Disbursement	50	10	7	7	5	0	8
Payroll	25	10	0	0	5	3	0
Wordprocessing	50	10	6	8	0	3	9
Fin. Modelling	25	10	0	7	0	8	9
Total[4]	400	400	265	292.5	162.5	135	340
OTHER EVALUATION[3]							
Personnel requ.	50	10	10	8	4	7	9
Training	50	10	10	8	6	7	10
Documentation	50	10	9	8	5	10	10
Vendor Stability	50	10	10	7	4	4	9
Total[4]	200	200	195	155	95	140	190
TOTAL POINTS	1000	1000	695	798.5	557.5	551	905
COST EFFECTIVENESS							
System Price	–	–	$24,100	$24,800	$23,400	$16,200	$30,500
Points/Dollar x 100			2.88	3.22	2.38	3.40	2.97

NOTES:

1. These items were evaluated subjectively on a 1-10 scale.
2. These items were evaluated on the basis of compliance with RFP specifications using a 1-10 scale, see worksheet for greater details.
3. These items were evaluated on basis of information included in proposals, company needs and public information using a 1-10 scale.
4. Total points awarded each system is the product of SYST.SCORE ÷ IDEAL SCORE X MAX. EVAL. PTS. for each evaluation item.

The first step is incorporated in the columns labelled **Max. Eval. Pts.** and **Ideal System.** The figures in the matrix are hypothetical but should indicate the relative importance of each item to the selection team.

The **Ideal System** data is based on the calculations done during the exercise on "Obtaining the right size System", with the application software specifications developed in the feasibility study.

In evaluating the viability of the project we looked at the areas that were feasible for the company to computerise, and the benefits that would be derived from automation.

Using the C.J.B. Ltd. example we will assume that the total benefit to the company from computerizing would be $12,000 per year.

The maximum amount the company will be willing to invest in the computer system may be assessed in a number of ways including Return on Investment (ROI), Internal Rate of Return (IRR), and Payback Period.

We will use payback period and make the simplifying assumption that all other costs of the project (aside from those included in the proposals) are negligible. If the company normally requires payback period of three years, then the maximum investment will be $12,000 x 3 years, or $36,000. As a result of this limitation, only proposals under $36,000 were evaluated. This would cover step two.

Step three is fulfilled in the last five columns of the matrix with data drawn from the proposals received from vendors. Proposals that did not meet certain minimum requirements (as spelled out in the RFP) were not evaluated. Also proposals that were out of the price range were not evaluated.

In considering step four, choosing our short list, we can see clearly from the **Cost Effectiveness** scoring on the Matrix that the Mylos 9000 system is lowest in over all points per dollar.

The only system with fewer overall **Total Points** (Moly 80A), was so much less expensive as to render it the highest scorer on the cost effectiveness ranking.

This points out the fact that we must be careful in how we use this cost effectiveness ranking. Even though the Moly 80A has scored highest in this ranking, it is very unlikely that it would be suitable for our needs because it scores so low in the Application Software section of the Matrix.

Having therefore eliminated these two systems, we are left with a short list of the MBI371, the System 5800, and the Hewlard 86.

Since the System 5800 has scored the highest of the remaining systems, should you

Apple III: worth a thought for a small business

By GORDON FINDLAY

Recently, I was able to spend several days with an Apple III, Apple Computer's small business machine. I had an enjoyable time putting it and some of its software, through their paces — and I was impressed with what I found.

It is a business machine, intended for small businesses or as a work-station in a larger network. A large RAM machine, it is supported by sophisticated operating software and powerful peripherals.

Hardware

In brief, the Apple III is a 6502B-based machine, with 128K of RAM memory, and a built-in mini-floppy disk drive. The 6502B is supported by additional hardware to enable bank-switching and extended addressing.

The whole unit is built around an aluminium casting which houses a large printed-circuit board and some smaller, "piggyback" boards.

On top of this low enclosure is the power-supply and built-in drive, with a low-profile keyboard at the other end. Both are in moulded plastic covers. The impression is of a quality product.

In photographs, the keyboard appears to be separate from the rest of the machine. But it isn't.

Four 50-pin expansion slots are found under the cover, between the power supply and the drive. At the rear of the machine are a number of connectors for the various interfaces provided and a monitor, or the profile hard-disk unit, sits neatly on top.

The 128K of RAM can be expanded to 256K, and a 4K ROM.

When the machine is turned on, a loader program in ROM thoroughly tests the hardware, then loads the operating system from the built-in drive. The ROM is then switched out. A monitor program in the ROM is accessible for advanced programming.

Keyboard

The keyboard has 74 keys, including a separate numeric keypad and cursor movement keys. Each key has auto-repeat, at about 11 hertz. This rate may be increased to about 33 hertz by holding down a special key, known as "Closed Apple" from the symbol on it!

The cursor motion keys are rather different — pressed normally they repeat at 11 Hz, but pressed more firmly, they repeat at the higher rate. This is very convenient when moving around the screen extensively, say in word-processing.

The general feel of the keyboard is quite positive — much better than the Apple II. Normally the keyboard has a 128 character type-ahead buffer.

All 128 ASCII characters are obtainable from the keyboard. A number of keys may be programmed for special functions.

The display supports upper and lower case, in three text modes — 80 x 24, black and white, normal and inverse; 40 x 24, black and white, normal and inverse; and 40 x 24, with 16 possible foreground and background colours.

All text modes have a software-definable 128 character set. The display also has four graphics modes.

Interfaces are provided for black and white, NTSC colour, and RGB colour video. Colours appear as a grey scale on black and white monitors. Interfaces are also provided for serial and "Silentype" printers for RS232 communications for audio output, and for two joysticks.

A built-in speaker is included, with provision for an external connection to a speaker or amplifier. This is supported with software to produce tones over seven octaves, with programmable duration and volume levels.

The standard monitor supplied is manufactured for Apple by Hitachi. It is a 12-inch, high-resolution green screen monitor, with a medium persistence phosphor, and has an anti-glare matt. While the monitor is perfectly adequate for business use, I have seen better.

Disk drives

As well as the built-in mini-floppy drive, a connector is available at the rear of the machine to connect an external floppy disk drive. This, in turn, has a connector to plug in yet another!

Of course, this "daisy-chaining" has to stop — in fact, the limit is four mini-floppies. Each has a capacity similar to the Apple II — 143-360 bytes. These drives, manufacturered for Apple by Shugart, are very quiet.

One of the Apple III's major attractions is its PROFILE hard-disk subsystem — a 5Mbyte (formatted) Winchester hard-disk assembly, with its own intelligent controller and power supply in a small unit, designed to fit between the Apple III and the monitor.

This is interfaced to the Apple through one of the expansion slots, and by installing its device driver into SOS, giving capacity of about 35 floppy-disks.

From page 14

automatically select it as your system? No!

Remember that your final selection should not take place until after you have talked with current users of the systems on the short list, run your test data through the systems, and are satisfied that a particular system is the one for you.

Now you do have most of the data needed to make an informed decision and are just about ready to make the final decision and implement the system of your choice. This is the topic of next month's article. See you then!

Operating system

The Apple III operating system is called SOS — "Sophisticated Operating System". The intention is that SOS handles all memory management, and all communication with the outside world, whether through keyboard, video, communications channels or whatever.

SOS supports a "heirarchical file structure". In such a structure, a directory of files on a disk is itself a file, and a directory may have other directories included within it. This means it is possible to have a file called, say /ACCOUNTS/RECEIVABLE/AUGUST/SMITH.

In this example, a disk is called ACCOUNTS, and RECEIVABLE is a sub-directory of ACCOUNTS, just as is AUGUST a sub-directory of RECEIVABLE. In this nest of directories, we find SMITH'S file — which contains data relating to accounts owing from Smith in the month of August.

This sort of directory structure is obviously intended to retain compatibility when a hard-disk is added to the system — where such a structure is needed to organise the large number of files on each disk.

Drivers

SOS communicates with hardware devices through "device drivers". These are special programs, parts of SOS, which handle all communication with the console (keyboard and display), printers, audio, communications hardware and graphics.

The use of device drivers has two main advantages — extendability and flexibility.

Closely allied to the device drivers is a program called SCP — System Configuration Program. SCP is used to add and delete drivers from the system, to tell SOS how many disk drives are in the system, to edit the character set displayed, and so on.

The main features of the device drivers for •CONSOLE, •GRAPHIX, •AUDIO, are mentioned in conjunction with the devices they control.

From the programmer's point of view, SOS handles all memory management for the system. Should you require it, SOS will provide the programmer with independent zero page and stack areas so that conflict between SOS and user software cannot occur.

Files are managed very efficiently. File access may be limited — for example, to read only, or read and write — and files may be date and time stamped automatically. The maximum size of size of a disk is 32Mbytes, and a user file 16 Mbytes. Of course, you need some fancy hardware to achieve these sizes!

Graphics

It was surprising to find this "business" machine has excellent graphics facilities. There are four graphics modes: 280 x 192 resolution, black and white only; 280 x 192 resolution, 16 colour with limitations; 140 x 192, full 16 colour; and 560 x 192, black and white.

In each mode, two pictures can be stored at once. Viewports may be set so that only part of the screen is altered, protecting images already on the screen.

In the colour mode 3, 16 colours can be freely mixed, for both foreground and background. And the interaction of these colours is freely alterable by software.

The principal languages for the Apple III are Business BASIC, and UCSD Pascal. Both are enhanced versions of the languages with numerous expansions to match the hardware capabilities.

Apple has chosen Pascal as the main system-development language, but BASIC is probably more familiar, and will be more likely to be used for local programming.

Business BASIC is an enhanced version of Applesoft, and will be familiar territory for most programmers. Enhancements have been made for better file handling. The long integer type (handling up to 19 digits) is included for financial calculations in which the normal integer type is too small, and the real type too imprecise.

A flexible PRINT USING statement makes for easy screen formatting, which is vital in business software. Program listings are automatically indented to show loops up clearly.

Modules provided by Apple include a VOLUMES module, to give information about what devices are in the system, and what disks are in the drives.

Turn to page 38

Microcomputer summary

NAME: Apple III.
MANUFACTURER: Apple Computer Ltd., U.S.A.
MICROPROCESSOR: 6502B, with extended addressing hardware.
CLOCK SPEED: Switches from 2MHz to 1MHz, average 1.4 MHz.
RAM: 128K.
ROM: 4K boot and diagnostics.
KEYBOARD: 74 keys (incl. numeric keypad), two-speed auto repeat, programmable function keys, cursor movement, alpha-lock.
DISPLAY: Three text modes (see text), upper and lower case; four hi-res graphics modes (see text); sixteen colours.
INTERFACES: Audio, B&W video, NTSC colour video, RGB video; RS232-C serial communications; serial printer; 'silentype' printer; two joysticks.
LANGUAGES: Business BASIC, Pascal, Forth, others.
SOUND: Built-in speaker, external output available; programmable tone generator, six bit DA converter.
DISK DRIVES: One mini-floppy built in, 140K formatted capacity; provision for up to three external mini-floppies, hard disks.
SIZE: Width 44.5cm, depth 46.3cm, height 12.2cm. Weight 12Ks.
PRICE: Basic unit, $6995.
REVIEW UNIT FROM: Computer South Ltd. Oxford Tce, Christchurch.

Computers' place on the farm

By C.R. McLeod

This is the first of a series of articles. We will start by looking at just what a computer can do to help you in the business of farming. When you hear about what a computer can do, what is usually meant is what software (programs) is available. Most computers can be programmed to carry out any specific job, it is the program which makes the difference.

A useful analogy is a tractor with several different implements. You have a tractor (computer) with no implements (programs). The result is that you cannot do anything useful with the tractor. If you buy a plough (one program), you can hook that to the tractor and the tractor then becomes useful. If you buy a grubber (another program), you can hook that to the tractor and use the tractor for a different task. This is the same as using the same computer to run different programs to do different things, for example, running a cash flow program then a games program.

One difference between the tractor/implement and the computer/program is that with the tractor and various implements, there are only a few ways you can join them (interface them). You can have a drawbar, or use three-point linkage. Admittedly there are three categories of three-point linkage, but almost everyone has standardised on category II. When an implement requires two-point linkage, you only use part of the existing three-point linkage, so there is no problem. This means that you can be almost certain that if you have a tractor with category II three-point linkage and a drawbar, any implement you might buy will fit.

With computers things are quite different. The linkage between the computer hardware (tractor) and the programs (implements) is far from standardised. If you have a TRS-80 computer, you will not be able to use a program written for an Apple (unless drastic, i.e. expensive, changes are made to the program or the computer).

The one operating system (linkage) which is anything like a standard and which is available on several different computers, is CP/M. There are also several other operating systems which can run programs written for the CP/M operating system, maybe with a few minor alterations. A couple of examples of these CP/M compatible operating systems are TURBODOS and MS-DOS. By using an operating system that is common to many different makes of computer, you are more likely to have a wider selection of programs available to you.

We seem to have got away from farming software, but it is essential that the importance of software, compared to hardware is understood. **Never** buy a computer before you know what software is available to run on the machine. In fact, look for the software first, then when you have decided what software you require, look for the computer which can run that software.

Now, to have a general look at what types of software are available. These can be divided into three major categories.

1) Financial planning and control.
2) Physical record storage and analysis.
3) Real-time date acquisition and control.

Probably the most common use computers are put to is in this area of financial planning and control. An important part of farm management is business management.

The first step is to prepare a budget before the beginning of the financial year. This is to determine what the likely income and expenditure will be. Once the budget has been prepared, a forecast cash flow can be estimated. Using the income and expenditure values from the budget for each section of income and expenditure (e.g. lamb sales, wool sales, expenditure on feed, expenditure on machinery), a month-by-month spread of income and expenditure is estimated. The result is a cash-flow forecast which gives you an idea of how much income and expenditure there will be, and when it occurs. This is what your bank manager likes to see at the beginning of the year, especially if you are talking of loans and overdrafts.

The next step is to update this information as the year progresses, using actual data. The actuals can be compared with budgeted figures (from the forecast cash flow) and hence a tight control can be maintained on the cash flow to stay within budget (if this is the intention). To maintain track of income and expenditure as it occurs, there are cash-book and accounting programs. Whereas the budget and forecast cash flow are usually only made at the beginning of the year, cash book and accounting programs are used throughout the year to constantly monitor actual income and expenditure. At the end of the year, the printouts from the cash book and accounting package can be handed to your accountant, which should make

Turn to page 27

From page 2

through central bulk buying. No mention of this to date. It is plain who loses by this, but who benefits? Standardising on one machine got New South Wales schools equipment at a price that left even special offers up in the air. Our schools get faced by punitive duties!

There is no sign of any coherent thought in the recommendations. They pass the buck to schools in every respect and most schools have no back-up to cope. Software and training support is stretched to deal with two machines, let alone five. Who will advise and assist with hardware maintenance?

It seems likely that the recommendations will result in several effects. The Apple will probably maintain a dominance simply because it has a large community of users who can help each other. Schools will feel it is a safe choice. The Poly will continue to attract support for its robustness excellent graphics and the BBC machine will gain devotes who can draw on the growing British and Australian educational software base. A strong Japanese presence seems unlikely at present.

The most important effect will be on educational opportunity and parity in education. The wealthy and well-organised schools will exploit this technology. More fortunate parents will simply outflank this emasculation of the school system. In some cases equipment will be bought and not fully integrated into school teaching due to poor support. In other cases the wrong equipment will be bought for the wrong reason. Over all, the significant opportunities available to widen our children's horizons will be missed.

If that seems to overstate the matter consider the policies adopted overseas. In Europe, America, and Australia educational computing is the subject of strong recommendations and meaningful support to get computers for education into ALL purchase, developing good software, disseminating it to others and supporting educators. It is a high priority even in times of recession. Have we in New Zealand concluded that they are all wrong and we are right? That computers in education are unimportant?

The recommendations to schools are a burning example of bureaucratic mismanagement. It has been said that as a nation, we are busy putting too much money into the wrong sort of silica. Instead of the silicon chip it goes into the mix for concrete to build dams and foundations of heavy industry. We are slipping behind in an enterprise that we have all the resources to excel in. There is an air of gloom in the complete lack of positive central response.

— Guest editorial by

Nick Smythe

U.K. Education Software for N.Z.

Six educational software packages have been released by Heinemann Publishers in New Zealand.

Developed in the United Kingdom the packages cost $35 each and run on the Apple II although they are also being modified to run on the B.B.C. computer.

The six packages are:

- Approximation, Estimation and Standard Form for mathematics and science students 11 years and older.
- Climate for geography students 14-16 years.
- Physiological Simulation for biology students 13-18 years.
- Scales for science and mathematics students 11 years and older.
- Transverse Waves and Longitudinal Waves (two separate packages) for physics and engineering students 14 years and older.

Heinemann's say they have set the price at $35 in an attempt to avoid piracy of the packages and the company says it will be "very firm" on the copyright situation.

B.B.C. Software

Teachers are being encouraged to write software for the B.B.C. computer.

Access Data, New Zealand wholesalers for the B.B.C., has set-up a subsidiary company called Educational Software to encourage the development of software for the computer.

Funded by sales of the B.B.C., Educational Software's two systems analysts and one consultant will act as advisors to schools to ensure the software is written to certain standards.

The software will then be offered for sale with the school that developed the idea receiving 25 per cent of the revenue.

In this way Access hope that programs peculiar to New Zealand educational requirements will soon become available. Software will also be imported from Australia and England, and Apple and other existing software modified to run on the B.B.C.

Prelude to a hardware war

This column is private opinion only. It is in no way representative of Education Department policy.

The official recommendation on computers in schools has finally been released amid a chorus of hoots and jeers. While the individual Australian states demonstrate just what a pack of stick-in-the-muds they are by supporting one or two computers at a time, New Zealand really showed a bit of flair by "standardising" to five.

While the support for Apple, Poly and the BBC Micro didn't surprise too many people, the presence of the two Japanese machines was sufficiently tangential to shake the early February blues out of the most depressed teacher. Anxious inquiries confirmed that no one had yet bought one, no one was considering buying one and several teachers thought they were video recorders included by mistake.

The whole task of recommending a particular computer for school use is particularly thankless. When the recommendation is almost accompanied by a statement that there is no money to fund school computers and nor is there likely to be any in the near future, it must depend on reasoned persuasion alone to give it any impact.

The names were nicely alphabetised, the diagram of the standard school desk was artistically executed, but the reasoned persuasion was nowhere to be found.

Enter that fabulous, never-say-die organisation, Polycorp.

Panache — Within about a week of the official recommendation, Polycorp released its own with a panache that already marks 1983 as a remarkable year. It sent a multi-page document to every school, promoting it as "everything you wanted to know about the fabulous five but couldn't find in the official statement."

Unlike previous mailings, which tended to be a bit stodgy, Poly's report was a rousing good read. It allocated points for different computer features with a cheerful inconsistency . . . a little comment here . . . a little comment there . . . and finished on a very high note by awarding itself first prize.

There is no denying that Poly is an excellent computer and with the appropriate long-term commitment a couple of years ago, New Zealand could have been on to a good thing. But Poly is in a Catch-22 situation. Enough schools have to buy to give it long-term viability; but schools are unwilling to buy until that long-term viability is demonstrated. The two inventors very publicly expressing their anger and frustration doesn't help things either.

Special offers — Despite advice to the contrary, most schools took up the Apple special offers in 1982. The Apple had

Turn to page 33

A taste of Australian honey

By SHAYNE DOYLE

The ranks of lower priced personal computers in New Zealand were swelled late last year with the arrival of the Australian-made Microbee.

The Z80 based MicroBee comes in four different options using a common case; 16K, 32K, and two 64K versions. Available user RAM is 14K, 30K and 56K, while ROM space is 28K of which 16K is the BASIC interpreter and 12K free for User applications. There is a separate 2K of video RAM, and 2K RAM for holding programmable character generator PCG characters.

By using low power 6116 CMOS RAMS, the machine offers a 'battery backup' feature, keeping power on the RAMS while the computer is switched off — great for those long chess matches, or rushing off across town to a friend's place to nut out a bug.

The MicroBee is a compact computer (13" x 10" x 2"), but when opened up there is still considerable space inside. The layout is a delight — the keyboard sits in the front half of the case. The rear section of the case, removable separately, houses two stacked circuit boards, filter capacitor, small speaker, and all board-mounted input/output sockets.

A 32K Microbee

The bottom board holds the main system circuitry and the interchangeable top (or 'core') board holds the RAM and ROM. Power supply is from a plugpack which runs fairly warm. All the important cables — cassette in and out, video out, power in — are on one DIN plug, and one thing I have noticed is the lack of TV interference, even though my television is only 10 feet away.

There are good I/O facilities — cassette program storage (Kansas City format), and data files at 300 or 1200 baud; 8 bit parallel port; fully buffered 50 pin Z80 expansion port; serial RS232C port at 300 or 1200 baud. While the 'Bee comes with the RS232 socket fitted as standard, one of my criticisms is that the parallel socket is offered as an optional extra — this also applies to the battery for the CMOS RAM backup — these should be standard.

MicroBee philosophy on using the various I/O 'channels' is delightfully simple — you just use normal Input and Print statements, "redirecting" the input/output to whichever channel you require. The manual has several pages explaining I/O Redirection and gives useful examples.

Display and graphics facilities are quite extensive for a machine in this price range although subject to limitations. The 16 line by 64 character upper and lower case display and low resolution graphics format of 48 x 128 six dot pixels is of course compatible with System 80 and TRS-80 programs, while high resolution offers 256 x 512 pixels.

The best graphics feature is the programmable character

BITS . . & . . . . BYTES . .

NEW ZEALAND'S PERSONAL COMPUTER MAGAZINE

PRESENTS:

The BITS & BYTES Computer Book Club

How it works —

Once you've bought a book, you're in the club. So just pick out the books you want, fill in the coupon on the next page and post it in.

What you get —

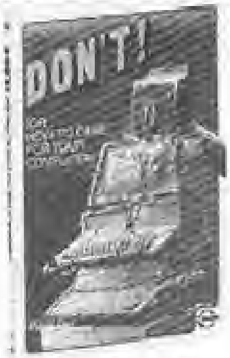
We are offering savings on the cash you pay for each book PLUS we give credit bonus points on each purchase. These are stored to your credit, and a number of times a year we will offer special great buys for cash plus these points. These bargains will be available only to those with credit points. You get the books post free!

This month's offer is only a beginning. We plan to widen the range of titles until we can offer one of the world's best ranges of computer books.

You've spent a lot of money on a computer. Make sure you get the most out of it by learning from these books how to make the most of it.

Just look at these prices! You get a special credit in addition! And they come right to your home by mail (We don't even charge for postage or packing)

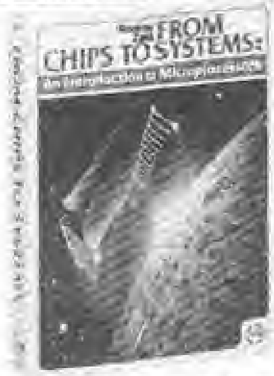
Look at these great computer books



Don't, or How to Care for your Computer — Rodney Zaks

An easy, entertaining guide to computer and peripheral preservation. Specific advice for the computer, floppy discs, hard discs, the CRT terminal, the printer, tape units, the computer room, software, and documentation. In the words of "People Computing" this book is "... cheap insurance".

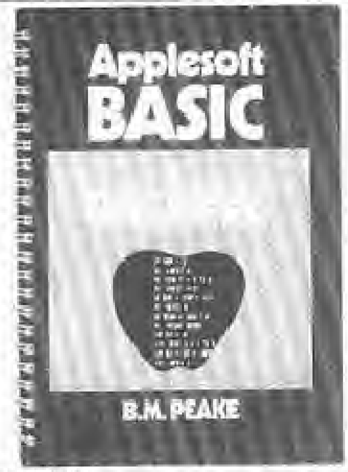
Sybex — Our price $22.75 **Save $1.20 and earn two bonus points**



From Chips to Systems: An Introduction to Microprocessors — *Rodney Zaks*

A superb, fast-paced journey through the history of microprocessors, the microprocessor chip itself, its support components, and the design of an actual microcomputer system. You'll find out that microprocessors were developed by accident rather than design, that early structural errors have become today's features and, most importantly, how easy it is to understand microcomputers. This is what "Elementary Electronics" said about this book: "One of the best introductory texts to the hardware end of things."

Sybex — Our price $30.35 **Save $1.60 and earn three bonus points**



Applesoft BASIC.
A Teach-Yourself Introduction — Barrie M. Peake

A manual for New Zealanders. It is reviewed in this issue of "Bits & Bytes". A one-book method of learning BASIC with the Apple, instead of picking information from two or three. It has model answers. Inquiries regarding terms for class sets welcomed.

John McIndoe — Our price $9.45 **Save 50c and earn a bonus point**

Mastering Computer Programming — P. E. Gosling

A complete, self-contained course from Britain for self study at home or for use in the classroom. In one book, the essential information to learn programming. The language taught is BASIC. The author, a former lecturer in computing, runs a computer services company.

Macmillan Master Series — Our price $9.45 **Save 50c Plus 1 bonus point**

Get the order in today!

Pick out your books, fill in the coupon, and get it in the mail today.
Stocks are not unlimited, and the last in may have to wait.

FREE POST TO YOUR HOME, PLUS — BONUS POINTS!

GET THIS IN THE MAIL TODAY:
To buy, cut out this coupon and post it to:

Freepost 125
"Bits & Bytes" Books
Box 827
Christchurch

If you address it in this way no stamp is needed!
Pay with order: Cheque, money order or fill in the Bankcard/Visacard boxes.

Payment:
SIGNATURE:
NAME:
ADDRESS:
(zone)

I enclose payment for the above order:

☐ Visacard
☐ Bankcard

Card holder signature
Date card expires
Cheque Cheque No.
Postal order ☐ Postal note ☐

Please note any areas of special interest:

Machine(s) in which you are specially interested

Please tick the appropriate boxes:
Please send me, post free, the following books:

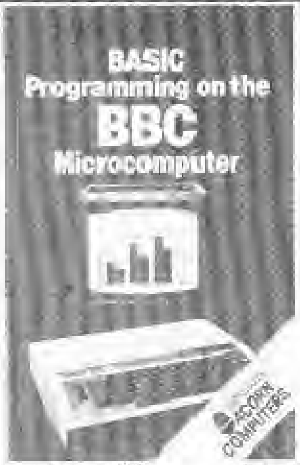
	Our Price	You Save	Bonus pts
☐ Quick Keyboarding (Alexander)	6.50	.45	1
☐ Introduction to T-Bug (Inman)	15.15	.80	2
☐ From Chips to Systems (Zaks)	30.35	1.60	3
☐ Inventory Management for Small Comp. (Atkinson)	24.65	1.35	2
☐ Introduction to TRS-80 Graphics (Inman)	18.95	1.10	2
☐ Basic BASIC-English Dictionary (Noonan)	20.80	1.15	2
☐ The Tenderfoot's Guide to Word Processing (Chirlian)	18.95	1.00	2
☐ Inside Basic Games (Mataosian)	25.18	1.32	2
☐ Don't, or How to Care for you Computer (Zaks)	22.75	1.20	2
☐ Applesoft BASIC (Peake)	9.45	.50	1
☐ Mastering Computer Programming (Gosling)	9.45	.50	1
☐ Mastering Computers (Wright)	9.45	.50	1
☐ Beginning BASIC (Gosling)	11.35	.60	1
☐ BASIC programming on the BBC (Cryer)	19.25	1.05	2
☐ Start with BASIC on the VIC 20 (Mount)	19.25	1.05	2
☐ Learning LOGO on the Apple II (McDougall)	18.05	1.00	2
☐ Atari Games and Recreations (Kohl et al)	27.00	1.45	2
☐ Basic Computing, A Complete Course (Crawford)	24.65	1.31	2
☐ WordStar made easy (Ettlin)	23.85	1.25	2
☐ VisiCalc: Home and Office Companion (Castlewitz)	31.96	1.66	3
☐ Discovering FORTH (Hogan)	29.60	1.60	3

Total payment enclosed

Read these beauts!

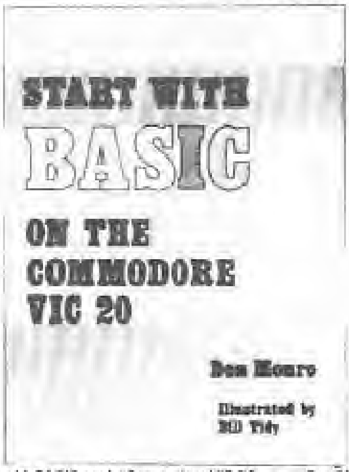
Mastering Computers G.G.L. Wright
A good introduction for the interested, or for use as a text in a computer appreciation course. Covers the field from the basis of mainframes, through such everyday applications as bar coding to the possibilities of videotex and an electronic money system. Well written in a concise, British style, and well illustrated.
Macmillan Master Series Our price $9.45 **Save 50c**
Plus 1 bonus point

Inventory Management for Small Computers Chuck Atkinson
A detailed inventory-management system, written in C BASIC and running under CP/M. Minimum hardware required: 32K RAM, and two disk drives. John Vargo said of this book in a "Bits & Bytes" review." . . . A good introduction to inventory management for a retail business, as well as a practical computerised solution to many of the problems presented in controlling this asset in a competitive environment.
Sybex Our price $24.65 **Save $1.30**
and earn two bonus points

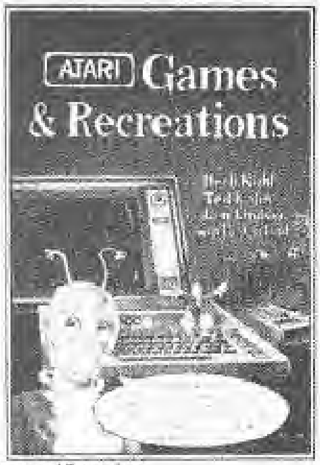


Basic Programming on the BBC Neil and Pat Cryer
You've seen the machine on television, and this is the book prepared to go with programme. It's designed for the new BBC. Teaches how to write programs, draw and animate pictures and graphics in full colour, design sound effects and program games. Detailed glossary.
Our price $19.25 **You save $1.05**
and earn two bonus points

Introduction to T-BUG Don Inman & Kurt Inman
This is the only book to describe in detail the machine-language monitor operations of the TRS-80. Each command is explained and discussed in detail and the examples are given to show how the commands may be used. Each step of every sample program is accompanied by a sketch of the corresponding video display.
dilithium Press Our price $15.15 **Save 80c plus**
2 bonus pts



Start with BASIC on the Commodore VIC 20 Don Monro
Don Monro is one of the snappiest, most humorous, and easiest to follow writers on beginning computing. This book with its illustrations by Bill Tidy, is an excellent guide for VIC 20 owners. The helpful exercises and line drawings make learning a snap.
Reston Our price $19.25 **You save $1.05**
and earn two bonus points



Atari Games and Recreations
By Herb Kohl, Ted Kahn, and Len Lindsay. Provides hours of pre-programmed games to play. But it also gives instruction readers need to improve on these games and create more complex and challenging games of their own. How to tell fortunes, compose songs, guess riddles, do word puzzles, and animate cartoon characters on the Atari.
Reston Our price $27.00 **You save $1.45**
and earn 2 bonus points

Quick Keyboarding
Sub-titled "Competent Keyboarding in 8 Hours", this book by New Zealander Vonnie Alexander has a unique method for teach-yourself competent keyboarding. A wall chart of finger position is included.
Methuen Our price $6.50 **You save 45c**
and earn one bonus point

Buy now! These are current prices only

BITS . . & . . .
. . BYTES . .

Introduction to TRS-80 Graphics — Don Inman

One for all the 80 users! A complete introduction to the basics of graphics programming on the TRS-80, using dozens of examples. It begins with the most basic concepts of line drawing and leads to geometric shapes, moving figure animation, and other more advanced topics.

dilithium Press — Our price $18.95. Save $1.10
and earn two bonus points

The Tenderfoot's Guide to Word Processing — Barbara Chirlain

Can I use a word processor in my profession or small business? What do I need in the way of equipment? Will a word processor be of use to me? These are just a few of the questions this book will answer.

dilithium Press — Our price $18.95. Save $1
and earn top bonus points

Basic BASIC – English Dictionary — Larry Noonan

Specifically for users of the Apple, PET, and TRS-80 who have found programs in magazines or books that were exactly what they wanted, but were written for some other computer.

dilithium Press — Our price $20.80 Save $1.15
and earn two bonus points

Beginning Basic — P.E. Gosling

An introduction to the language for first-time users. Also has lots of interesting digressions about such devices as tape readers.

Macmillan — Our price $11.35 You save 60c
and earn 1 bonus point

Inside Basic Games — Richard Mateosian

A fun introduction for anyone beginning programming. The author has chosen the medium of games to teach readers how to design error-free, interactive BASIC programs. Computer games are described in detail, then explained and analysed to illustrate how the games were developed. Rules, algorithms, and coding differences from the TRS-80, Apple II, and PET are also included.

Sybex — Our price $25.18 Save $1.32
and earn two bonus points

Basic Computing: A Complete Course — Tim Crawford

From McGraw Hill (Canada). Provides a broad-based introduction to computer science and data processing, suitable for a variety of levels in high schools, universities or industry. Begins with questions such as What is a Computer? Where Did Computers Come From? Gives detailed coverage of computer languages, programming, program structure, logic, testing, documentation and program maintenance.

McGraw-Hill — Our price $24.65 Save $1.31
and earn two bonus points

WordStar Made Easy — Walter Ettlin

In 14 easy lessons, this handbook demonstrates all the powerful features of the MicroPro WordStar system. Lessons cover everything from loading, using the unique printing, text manipulation, and formatting functions. Spiral bound, it is ideal to use while working at the terminal.

Osborne — Our price $23.85 You save $1.25
and earn two bonus points

VisiCalc: Home and Office Companion — David Castlewitz, et al

Provides extensive coverage of the popular VisiCalc program, and can be used by the novice and expert alike. For the novice, it offers 40 models that can be used immediately for personal and business applications. For the expert it is a source of new ideas and techniques.

Models created with Apple 3.2 version, but will perform just as well on machines with other versions.

Our price $31.93 Save $1.65
and earn those bonus points

Discover FORTH: Learning and Programming the FORTH Language — Thom Hogan

Whether you are a beginner seeking information on this multi-faceted programming language or a serious programmer already using FORTH this book is a reference that should not be overlooked. Describe FORTH syntax, specifically applicable to both FORTH 79 and FIGFORTH.

Our price $29.90 You save $1.60
and earn three bonus points.

Learning LOGO in the Apple II — McDougall, et al

LOGO's a Piaget-based way into computing. A multi-purpose language. Non-technical, learning by doing.

Prentice-Hall (Australia) — Our price $18.05
You save $1.00
and earn two bonus points.

POST YOUR SUBSCRIPTION TODAY!!

NAME: ....

ADDRESS: ....

....

.... ZONE: ....

I enclose payment for one year's subscription to BITS & BYTES

Adult $8 ☐ School $6 ☐ School name ....

Form ....

PAYMENT:

Visacard
Bankcard ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Card Holder Signature ....
Date Card Expires ....
Cheque ☐ Cheque No. ....
Postal order ☐ Postal note ☐

6

Please answer the following questions:

My interest is:
Farming
Education
Competitions
Business
Professional
Hobby use
General news
Programs
Beginner

My field is:
Professional (medical, accounting etc)
Business
Education (lecturer, teacher etc)
Consumer (hobbyist etc)
Farming
Student
Other

Do you use a microcomputer Yes/No

If yes, which brand ....

BACK COPIES

Please send me the following back issues at $1 each plus 50c postage and packaging each.

Issue No(s): _ _ _ _ _ _ _ _ _ _ _

Number of Copies: _ _ _ _ _ _ _ _

(Please send payment with order. Note; no more copies of Issue No. 1 or 2 are available and other stocks are limited)

No stamp is required if you write this address on your envelope

Freepost No. 125
BITS & BYTES
P.O. Box 827
Christchurch.

generator (PCG), which allows you to design up to 128 custom characters, each made up of 16 x 8 dots. Very detailed images and pictures can be built up using groups of these PCG characters and displayed anywhere on the screen. The statements SET, RESET, POINT, PLOT apply to both low and high resolution graphics. Other display options are Inverse video and Underlined text.

While text may be mixed with graphics, the five modes INVERSE, UNDERLINE, HIRES, LORES, PCG may only be exclusively selected and this can be frustrating at times. There are ways around this however, you can simulate HIRES lines with PCG characters for example. A major limitation to Hires graphics lies in the method of using the PCG RAM for Hires points — a tradeoff between high graphics memory requirement and high resolution. It means you can't draw those amazing line graphics the more expensive machines can do, as you are limited to 128 PCG or Hires characters, but it does allow you an introduction to the field.

This month will see a colour option released in Australia, offering 26 foreground and 26 background colours, but the drawback here is that colour output is to an RGB monitor or specially modifed colour television set and not via a modulator. Normal B&W display can be fed through a modulator to a domestic TV set. MicroBee has an on-board sound generator and speaker capable of producing two octaves from A below middle C.

The BASIC interpreter is a custom product called "Microworld Basic", and those moving up from a Super 80 will recognise it. Programmers brought up on a diet of Microsoft or 'S-80' Basics will probably not take to it. You need to be more disciplined in your approach to writing software — in itself a very good point, as it only allows variable names A – Z, A0 thru A7 to Z0 thru Z7, and great care needs to be taken when mixing real and integer variables. Two functions are provided to convert from real to integer and back, when the inevitable "mixed mode error" occurs.

String handling follows the ANSI "string slicing" convention whereby A1$(;5,12) gives you characters 5 to 12 of the string A1$ — a welcome relief from LEFT$ and its cronies. A superb feature you will love is the search and replace command GX — the command GX/A1/Z7/ will stop at every line containing A1 and give you the option of having it changed or leaving it. You have to use this a few times to appreciate the power and convenience it offers.

All the usual BASIC commands and functions are there in one form or another, and the interpreter supports parameters in subroutine calls, seven user defined functions, powerful renumbering (you can renumber segments of a program) and auto line numbering, and once you gain a bit of experience I am sure you will come to like this BASIC.

MicroBee has networking facilities which must appeal to the school market, and I know of three local schools who have purchased several machines each with a view to implementing this.

The 64K versions offer two routes to disk expansion — S100 CP/M disk system, or a cheaper alternative using an on-board disk controller and a powerful DOS which among other features tests the format of a disk loaded in and boots itself accordingly. Both disk versions have a 24 x 80 characters display.

Other goodies available are an Editor Assembler in ROM (EDASM), Word Processor in ROM (WordProc), Wordstar word processor on CP/M versions, ROM Chess, and an invaluable machine code programmers utility toolbox program which has been developed locally by two users — BEEMON. This allows you to

Turn to page 23

Microcomputer Summary

Name:	Applied Technology MicroBee, Models 16, 32, 64
Microprocessor:	Z80, 2MHz for models 16 & 32.
RAM:	16K, 32K, 64K. Video RAM 2K. PCG RAM 2K.
User RAM:	14K, 30K, 56K.
ROM:	16K BASIC ROM, 12K User ROM.
Input/Output:	Cassette interface at 300 or 1200 baud, RS232C serial — 300 or 1200 baud, 8 bit parallel port, 50 pin Z80 expansion bus, Video.
Keyboard:	Full size 63 key, auto repeat on all keys.
Display:	16K & 32K — 16 lines by 64 characters, upper/lower case, inverse video, underlining. 64K — 24 lines by 80 characters.
Languages:	MicroWorld BASIC, Z80 machine code.
Graphics	Low resolution — 48 x 128 6 bit pixels. High resolution — 256 x 512 pixels. Programmable characters — 128, 16 x 8 bit characters. Colour — 26 foreground, 26 background, RGB ouput.
Sound:	Monotonic sound generator & speaker, 2 octaves.
Costs:	16K — $829. Upgrades to 32K — $161, to 64K — $561.
Options:	Parallel port, battery backup for CMOS RAM, Colour board.
Peripherals:	Disks — S100 disk system requires Expansion unit, disk controller, 2 drives CP/M DOS. Or MicroBee disk system, requires replacement core board with on-board disk controller and 1 drive.

In the Belly of the Beast: 4

A bit more about binary

By GERRIT BAHLMAN

When a group of bits is used to represent a number, the bits starting from the right-hand end are used to indicate the number of 1s, 2s, 4s, 8s, etc, in the number. I showed you how we convert numbers into this form in the second article on the binary number system. Therefore, if five bits are used to write down a number in binary, say: abcde then the number would be: $a \times 2^4 + b \times 2^3 + c \times 2^2 + d \times 2^1 + e \times 2^0$

The largest number that could be written like this would be 11111, which, if you work it out, is

$16 + 8 + 4 + 2 + 1 = 31.$

The smallest value would be 00000 = 0.

But this would only allow us to write positive numbers. How could we write negative numbers? What is the difference between positive and negative numbers? Well, obviously only the sign. So, if we store the sign somewhere in the number, either a positive or negative, then we could always tell what sort of number it was. This is almost what is done. But with one slight improvement. We don't need to remember that a number is positive. Only that it is negative. So, instead of saving the positive sign we could save one value of the computer word and just save the negative. This is how that is done.

We will use the five-bit example to explain how it works. Since we want to represent negatives it would seem sensible to divide the range of numbers from 0 to 31, (32 in all) into a positive and negative half. The first bit is a "1" if the number is negative and a "0" if it is positive. Let us look at a number and see how it could be written.

Positive 5 would be written:
0 0 1 0 1 in binary.

Negative 5 would be written:
1 1 0 1 1 8
−16 + 8 + 2 + 1 = −5
in binary

Notice that our negative number is written as a sum of a negative and a positive number. The first bit, called the most significant bit is worth −16 while the rest of the bits make up the positive amount which will leave −5. It sounds awfully complicated but in fact it's quite easy to do. Let's look at the number 5 in binary again. We will still only use five bits.

0 0 1 0 1

Swop all the 0s and 1s.

1 1 0 1 0 (This is called the complement.) We almost have the negative 5 that we want! What is the difference?

Add one to the complement.

1 1 0 1 1

Now we have −5.

This system works for any number provided that it fits into the number of bits available to it. Obviously, 16 bits would allow a much larger range of numbers to be represented, 32 bits even more, etc. Incidentally, what would the range of positive and negative numbers be for a machine which used 16 bits of memory to store a number? What would 1 0 0 0 0 mean in this numbering system? −16? or −0?

The second may seem strange but there is a good reason why some machines think of it as negative zero. If a programmer tried to store a number which was outside the range of the machine's capacity, (the number of bits available to store a number), then the machine should tell the programmer.

In particular, if the arithmetic resulted in a large positive number suddenly becoming zero you can be fairly sure someone would notice. At any rate, some machines would call the number −16 others −0 but it doesn't matter too much which because both numbers are at the limit of the machine's representation limits.

This method of number representation is common to most machines, but, of course, doesn't allow for fractional numbers. A variety of methods is used for these and we may take some space in future to look at a few of these.

In addition to representing numbers, words are often used to represent non-numeric information such as customer names and addresses, standard letter formats, telephone directories, debtor and creditor lists, car registrations, etc. You ought to realise that more computer activity is concerned with non-numeric information than numeric information. In other words we spend more time "remembering" written information than doing calculations!

In article 2 we looked at the way in which non-numeric information could be represented using "bit patterns". Each letter, digit, punctuation mark, etc. is represented by a different bit pattern. The number of bits needed to represent a character is determined by the number of different characters. Remember that N bits can represent up to 2 characters.

This little table shows how many characters can be represented by various numbers of bits.

N	MAXIMUM NUMBER OF CHARACTERS	NAME OF SYSTEM
6	64	Binary Coded Decimal
7	128	
8	256	American Standard Code for Information Interchange, Extended BCD Interchange Code.

The 7-bit system turns out to be a little awkward so the 8-bit system is used. Two forms of it are found, the more common in the home computer market being ASCII. The 6-bit system isn't really big enough to allow for special characters such as lower case and control characters used to send special messages to various peripheral devices. The ASCII code actually has several dialects depending on the brand of the machine but EBCDIC is pretty well the sole property of IBM and so no other versions of it have appeared. The ASCII code is sometimes altered to provide a parity check bit. This means that a special check is incorporated in the code to allow for accuracy of the character being checked.

Parity checking

Parity checks involve adding 1 to the bit pattern, if necessary, to make the total number of 1's even or odd. If a mistake occurs and one of the bits is wrongly read or written, the machinery will notice that the odd or even property is not obeyed and it will register a parity error. When the even number of 1s is used the machinery is said to register even parity. Otherwise, the machine is said to register odd parity. Let us invent a character to be sent using eight bits.

1 0 1 0 1 0 1 1

We will assume that we want even parity. Before we send this character and save it in memory or write it on the typewriter we must add a 1, or put an extra bit on the character to make the number of 1s even. Let us assume that we will always adjust the second most significant bit so that there will be even parity. So, we actually send:

1 1 1 0 1 0 1 1 — six 1's and six is even. Or, if it were odd parity that we were going for:

1 0 1 0 1 0 1 1 — 5 ones, we would not need to change anything!

There is of course one fairly important assumption that we have made. We hope that there will only be one error in the character to be sent. What would happen if two bits were mis-read or mis-written?

EBCDIC has one advantage over ASCII. The binary codes for the characters are in alphabetical order. If a character is alphabetically after another its code in binary is larger. So, if you want to sort or collate stored information you do not have to convert it into its character form, you just read the size of the binary number! This is very useful from a speed point of view when large amounts of written information are to be rearranged.

In the next article we will look at addressing memory.

From page 21

virtually wander at will through MicroBee's memory, editing, displaying memory contents in ASCII/Hex/Symbolic format, and many other features are incorporated in this single ROM.

The program base is growing rapidly, and much of the software is written by users themselves. Needless to say, it includes a superb version of Space Invaders.

In summary, MicroBee has a few bad points — cheapish "clacky" keyboard, optional parallel port socket and memory backup battery, HIRES graphics limitations, only average sound facilities. MicroBee's good features definitely outweigh these however, and I can say that I would still buy the machine having used it for six months now.

A 16K machine will cost you $829, 32K is $990, 64K cost $1390. S100 disk expansion will cost you $3350 for expansion unit, disk controller, 2 drives and CP/M. The three ROM based programs EDASM, WordProc and Chess cost $90, $150 and $50 respectively. Most cassette programs are $14. BEEMON costs $25 and is obtainable from J. Durham, 16 Hudson Avenue, Upper Hutt.

Additional VIC Dealer For Competition

Bay Office Supplies
87 1st Avenue
TAURANGA
Phone 81-009

Exponential numbers

Basic BASIC 5

By Gordon Findlay

Continuing a series on BASIC for complete beginners.

This month, some mathematics — simple stuff — two more statements to add to your arsenal, and an answer to a correspondent.

Computers are not the sole preserve of mathematicians. That should be pretty obvious to all of you who are not mathematicians! But a certain amount of elementary — and I stress the word elementary — mathematical information is an advantage in programming. After all, even when a computer program is manipulating names and addresses in a mailing list, the operations involved are arithmetical.

An aside. Computer programmers have not always been drawn from the ranks of mathematicians. In the really early days (mid-1950's) of computers, one of the pioneers in the Cambridge University computing laboratory suggested that the best people to recruit were those who had specialised in languages, or classics. (He also suggested that women made better programmers than men, but we need not dwell on that!)

So here goes. First set this program running. All it does is print a table of numbers, doubling at each stage.

```
10 X = 2
20 J = 1
30 X = X * 2
40 PRINT X
50 J=J +1
60 IF J V 40 THEN GOTO 30
70 END
```

At the risk of repeating myself, don't forget to put the LET in statements 10, 20, 30, and 50 if your machine requires it.

Now everything here starts off fine, printing the numbers 2, 4, 8, 16, and so on. But then what happens? Numbers start appearing in a different format. The last number printed by the program above when I tried it was 1.09951163E+12. This manner of writing numbers is called STANDARD form, EXPONENTIAL notation, or (rather incorrectly) SCIENTIFIC notation. The symbol "E" means "times 10 to the power of", so the number I wrote above is read "1.09951163 times 10 to the power of 12". The best way to think of this "times 10 to the power" business is to regard it as an instruction to shift the decimal point. So to convert that number to the usual form, shift the decimal point over 12 places (to the right): 1.09951163E+12 = 1099511630000. Here are some others to check:

2.7135E+3 = 2713.5
9.87654321E+5 = 9876543.21
2.987654E+0 = 2.987654 (moving zero places of course).

Now change the program above — replace line with :

```
30 X = X/2
```

So the numbers get smaller instead. The result? I got 3.63797881E−12. Notice that the exponent (the power of 10) is negative this time. What this means is that the decimal point must be moved to the LEFT. So the result this time is 0.00000 00000 03677, etc. There are 11 zeros after the decimal point. Here's your homework — check these!

3.1415E−4 = 0.00031415
1.2345E−1 = 0.12345

What use is this esoteric way of writing numbers? The main advantage is that very large, and very small numbers can be written in a compact form, which is easy to read. The disadvantage is that there is a limit to the number of "significant digits" that your computer can handle. That first answer which we got, 1.09951163E+12, is only an approximation. The exact answer is 109 95116 27776, and you can see that the answer is a fairly good approximation (I had to get that exact answer manually). So, to sum up, the use of standard form allows us to use very large, and very small, numbers, with a small loss of accuracy.

Numbers in standard form can be used in programs as well as being output. Here is a simple program, which calculates (roughly) the volume of an electron.

```
10 RA = 2.81777E-15
20 REM RA = RADIUS OF ELECTRON IN METRES
30 PI = 3.1416
40 REM VL WILL BE VOLUME OF ELECTRON.
50 VL = 4/3 * PI * RA * RA * RA
60 PRINT "VOLUME OF ELECTRON IS:"; VL; "CUBIC METRES"
70 END
```

An exercise for you: use standard form to write a program which finds the total distance travelled in 6 years by 2 million cars, if each travels 20 000 km per year.

Order of operations

Just to remind you, the order in which mathematical operations is done is not a free choice. For example, in the statement:

```
10 X = 1 + 2*3
```

The result is X = 7, not 9! The multiplication must be done first, rather than adding first. In general, multiplications and divisions are done before additions and subtractions. The order can be altered by using brackets: X = (1 + 2) * 3 does give 9.

READ and DATA

Frequently a program needs to have a whole lot of numbers every time it runs. If the program always uses the same values, it is silly to have all these numbers input by

the program user each time — better to build them in. But assignment statements may not be the best way of getting several values in. There is an alternative — READ and DATA.

Here is a contrived example:

For some reason, we need to do date calculations in a program. One standard way to handle our rather irregular calendar is to store the number of days in each month in advance. This could be done by 12 statements like this:

```
10 JAN = 31
20 FEB = 28
30 MCH = 31
40 APR = 30
.
.
.
120 DEC = 31
```

This long list takes time to type, and takes up space. These numbers are never going to change, so it would be ridiculous to use INPUT statements and put them in through the keyboard every time the program runs. There is a better way than any of these! The READ statement picks up values for variables from a list, given in a DATA statement. The 12 statements above could be collapsed in two:

```
10 READ — JAN, FEB, MCH, APR, MAY, JUN, JLY, AUG, SEP, OCT, NOV, DEC 20 DATA 31, 28, 31, 30, 31,30, 31, 31, 30, 31, 30, 31
```

The READ picks up the items in the data list and gives the value to the matching variable. The first value in the data list goes with the first variable in the READ list, and so on.

The first time a program encounters a READ, the first DATA statement is used, the next READ uses the next DATA and so on similarly. If the DATA statement has more values than are read, the rest will be ignored — although a later READ might pick them up. The program will stop, with an error message, if there are more variables than values.

The program relates READ-DATA pairs by the order they appear in the program, so the DATA statement doesn't need to be next to the READ. This can be useful if you have a lot of these statements, as all the data lists can be put together at the end of the program. This usually makes it easier to read.

I had a letter the other day which contained a question of general interest:

> "What is the difference between the two BASIC commands STOP and END? I have been trying some programs on the Apple and they both seem to do the same — stop the program dead!"

Well, there isn't any real difference. With most machines, the STOP statement produces a message saying where the program was when it stopped, but the END statement just stops and doesn't produce any special messages. On the Apple, a STOP statement gives a beep and a message like "BREAK IN 100" meaning in program line 100. This can be useful in debugging, or in cases in which you intend to CONT(inue) after the break.

I have been asked repeatedly to recommend a book for learning BASIC. There are literally hundreds, and I haven't looked at many of them, so I can only mention two that I have seen which looked good.

"Instant Freeze-Dried BASIC", — written by an author with a wild sense of humour, and snakes crawling through programs.

"A Bit of BASIC" — more conventional, with specific sections for Apple, TRS80 and their graphics.

Banish those memory expansion blues

By PETER ARCHER

Most new VIC users are probably confused by the apparent complexity of memory expansion options for their machine, and even more by the fact the location of the screen memory moves when memory is expanded in some ways. They may have found that some programs written on an unexpanded VIC will not run properly on a memory expanded machine.

In this article I will try to explain how and why this screen memory movement takes place, and detail a simple way of writing or re-writing your VIC programs to allow for this movement.

Figure 1 shows three versions of a memory map of the main RAM area of the VIC. These are: the standard unexpanded VIC; the VIC expanded by an extra 3K bytes; and expansion by up to an extra 24K bytes.

Unexpanded VIC

There is a total of 5K bytes of RAM equipped within the bottom 8K area of the VIC's memory area.

The bottom 1K of memory from address decimal zero to 1023 is used by the machine itself for system variables, cassette buffer, etc. Most detail of this area is outside the scope of this article.

From decimal 1024 to 4095, there is a 3K empty gap. (Note I am using only normal decimal numbers, as I believe the use of "hex" numbers can confuse beginners and is not essential to grasp the concepts I will deal with).

There is 4K bytes of RAM equipped from 4096 to 8191. However, the top one half K of this area (from 7680 to 8191) is used for screen memory storage and is not available for user programming. This leaves the area from address 4096 to 7679 free for BASIC and/or machine language programming, and is the "3583 BYTES FREE" referred to on the screen message when VIC is first powered up.

The area from address 8192 to 32767 is empty and is available for RAM (and/or ROM) expansion via the memory expansion port at the right rear of the machine. RAM is usually expanded in a minimum of blocks of 8K, although it can be done in 2K blocks.

Expansions

The extra 3K bytes of RAM fills the gap in the standard machine from 1024 to 4095. This results in 6.5K of continuous RAM available to the user from 1024 through 7679 which can be used for BASIC and/or machine code programming. The screen memory stays at the same place (7680 to 8191) at the top of this area, as before.

The area from 8192 right up to 32767 can have various combinations of RAM and/or ROM added. ROM cartridges are available both from Commodore and from independent suppliers (e.g. "toolkit" ROMs, machine language monitors, etc.).

If RAM added in this area is to be available for programming in BASIC, it must be "continuous" with existing internal RAM from address 8192 upwards.

Any gaps in this RAM will mean the VIC operating system cannot "see" the RAM above the gap, and it will be ignored by the system which allocates memory space for storing BASIC programs. (Any RAM above a gap would still be available for machine language programming, however).

Reference to the memory map of the standard VIC in Figure 1 will show that screen memory normally resides from 7680 to 8191 - between the 3.5K of standard internal RAM and any RAM which is added from 8192 upwards.

"Break"

To overcome this potential "break" in the continuous BASIC RAM, the VIC operating system automatically relocates the screen memory to the bottom of the internal RAM, from 4096 to 4607.

User available BASIC memory now starts at address 4608 and extends upwards through 8192 to the top of whatever RAM is added continuously to a maximum of 32767. The maximum of RAM which can be available for BASIC is therefore 32767 minus 4607, i.e. 28160 bytes or 27.5K.

But what about the 3K below 4096? Can this area still be used? Yes, that 3K gap can still be filled with RAM, but this 3K of RAM will not be available for use by BASIC, as the screen memory breaks it off from the rest of the RAM. It can still be used for machine code programming or other special purposes.

Where, you may ask, does this leave the poor programmer who writes a program which includes poking of characters into the screen memory, and wants the program to be able to run on both a standard (or 3K expanded) VIC and a VIC expanded above 8192?

Answer

The answer is to define the start of screen memory as a variable, the value of which is set by the machine when the program runs. This screen start variable will have to be set by the VIC to a value of 7680 for an unexpanded (or 3K expanded) VIC and to 4096 for a VIC with the screen memory relocated because of expanded memory.

How do you achieve this? Memory location 648 contains the information required.

Peeking into address 648 on a standard VIC, or on one with only the bottom 3K of RAM expanded, you will see the number 30. Peeking into 648 on a VIC expanded above address 8192, you will see the number 16.

A little bit of simple maths will tell you that 30 multiplied by 256 equals 7680; and that 16 multiplied by 256 equals 4096. If you define your screen, start variable as "256 * PEEK (648)".

Unfortunately, this is not the end of the story because while all this is going on, other things are happening in VIC's higher memory

area. As a colour computer, the VIC not only requires a screen memory but also a screen colour memory to set the colour of the screen characters.

In a VIC with the screen character memory located starting at 7680, the colour memory is located in a ½K block of RAM between 38400 and 38911. If the screen character memory changes locations to 4096, the colour memory shifts down to a different ½K block of RAM between 37888 and 38399.

Routine

A suitable initialisation routine to take care of both these points is:

```
10 SS = 256*PEEK(648) : CS
   = 38400
20 IF SS = 4096 THEN CS =
   37888
```

Pokes to the screen can then be done this way:

```
100 POKE SS + C + 22*L,A :
    POKE CS + C + 22*L,B
```

Where:

SS is the "screen start" location
CS is the "colour start" location
C is the column number (0 to 21) of the screen grid
L is the line number (0 to 22) of the screen grid
A is the screen code number of the required character
B is the code number of the colour required

Screen code and colour numbers are listed in Appendices H and I of the VIC user manual.

From page 17

his job considerably easier (and cheaper).

These programs will not save a great deal of time in collecting information. Time is saved in preparing the results (output), not in feeding in the information (input). Once the information has been entered, however, providing the programs are well written (more on that later), much more information should be available than if it had been processed manually. This is because comparisons between budget and actual figures can be made easily and quickly, and such things as trial balances can be quickly obtained.

Physical record storage

We generally distinguish between financial records and physical records. The storage and analysis of financial records has been discussed above. This section covers physical records.

A physical record refers to an item of information about quantity (or quality) on the farm. It might be the number of hoggets on the farm, a wheat yield, a wool weight or wool count, a sheep weight, or a daily rainfall.

In general terms, physical record storage and analysis involves recording various items of information, then analysing this information to obtain results which will be useful in making management decisions.

Real-time data acquisition and control

The third category of software is real-time data acquisition and control. In many cases, this is related to physical record storage and control.

To define this title I will start at the back. The "control" part is just as it says, the computer is issued to control certain events (e.g. turn irrigation sprinklers on and off). The "data acquisition" refers to the collection of data by the computer from the world around us (e.g. read and store the temperature every five minutes). The "real-time" means that the computer is collecting data or controlling some event, as that event occurs. Instead of someone reading the thermometer, then keying the temperature and time at which the reading was taken into the computer, the computer is used to read the temperature directly as it happens. This is what robots do. They sense what is going on around them (e.g. car body moves into place) and then control some activity which is appropriate at that time (e.g. spot weld a join in the car body).

The potential for real time data acquisition and control on the farm is considerable. We will cover more of this in later articles.

Space Invaders

This program, by D. LOVATT and S. LOVATT, of Gisborne, will fit into the 1K ZX81 or into the 16K ZX Spectrum (with some additions). The program, although small, is divided into several sections:

1: Lines 10-70
   This section contains the initialisation of variables for the program. The invaders are stored in the two-dimensional array a$ which is dimensioned in line 10. I used CODE and VAL extensively to save memory. A number like "5" takes up six bytes in memory, while an expression like "VAL"5"" takes up 4 bytes. Even more memory can be saved by using "INT PI" for 3, "NOT PI" for 0, and "SNG PI" for 1.

2: Lines 80-220
   This section contains the actual game itself. The statements in lines 110, 150 and 160 may seem unusual to some. The statements, when executed by the computer, give boolean results (that is, 1 for true and 0 for false). If you now read the statements using the conditions mentioned, they will make sense. The PEEK statement in line 130 just peeks the printing position on the screen. Line 120 positions the peeking position.

3: Line 230
   Prints the score at the end of the game. Note that you only have to destroy the top two rows of invaders to end the game.

Computers with a larger memory (such as the expanded ZX81 or ZX Spectrum) can expand the program by an incredible amount. Take note, however, that the expanded ZX81 is considerably slower, so you will need to lower RAMTOP to speed it up.

HOW TO PLAY:

Use inkey$ "5" and "8" to move left and right respectively. "0" to fire.

Listing:

```
10   DIM a$(3,5)
20   LET a$(1)="AAAAA"
30   LET a$(2)="AAAAA"
40   LET a$(3)="AAAAA"
50   LET s=0
60   LET n=CODE "="
70   LET x=CODE "?"
80   FOR h=INT PI TO CODE"?" STEP VAL "2"
90   FOR v=VAL "5" TO CODE "$"
100  PRINT AT h,v;a$(1);TAB v;a$(2);TAB v;a$(3)
110  LET x=x+(INKEY$="8")-(INKEY$="5")
120  PRINT AT n,x;
130  IF PEEK (PEEK 16398+256*PEEK 16399)=38 THEN LET
     a$((n-h)+1,(x-v)+1)="△"
140  PRINT"*"
150  LET n=n-(n<20)+(22-h)*(n<h)
160  LET n=n-(INKEY$="0")
170  LET s=s+1
180  CLS
190  IF a$(1)="     " AND a$(2)=a$(1) THEN GOTO 230
200  NEXT v
210  NEXT h
220  GOTO 60
230  PRINT 1000-s
```

△= space

Commandos short-cut

A reader has suggested an amendment to C.W. Nighy's BASIC game for the hand-held Casio FX702P, "Commandos", printed in the December-January issue of BITS & BYTES. Mark Wilson, of Christchurch, suggests that lines 80 to 119 be replaced by:

```
75 J=1
80 IF C$=MID (J,I) THEN 83
81 J=J+1:IF J=11 THEN 70
82 GOTO 80
83 IF A=J-1 THEN 200
84 GOTO 210
```

Line 3 is then re-written as

```
3 VAC: $="0123456789"
```

This gives a much shorter version of the game. Mr Wilson writes of this latter portion of his amendment:

"This program segment is quite a useful one: it scans the special character variable $ and returns a numerical value co-responding to a character as shown. It also recovers if the input (C$) is not '0'. . .'9'."

He adds that there are redundancies in many of the listings in the FX-702P's program library.

Bringing Computers to life

CompüteX'83

MAY 27-29, 1983
CONVENTION CENTRE
PALMERSTON NORTH

RADIO MANAWATU 2ZA

Personal Computers ★ Home Computing ★ Small Business systems

In all walks of life, personal computers have revolutionised computer power - bringing it within the reach of a far wider and more popular market than ever before.

In 1983 this revolution will explode into the home and personal computer market. To meet this demand Radio 2ZA, Palmerston North announce a brand new event - Computex '83. The promotion of the exhibition will be heavily geared to attract the growing market of potential and existing users of personal computers, from home computer enthusiasts to businessmen.

Bring your computers and services to life - all walks of life - at Computex '83.

Complete and return this coupon today and we'll send you the details.

Exhibition Manager,
Computex '83,
Radio 2ZA,
P.O. Box 1045,
Palmerston North.

CompüteX'83

Name ....

Position in Company ....

Address ....

....

2ZA544 .... Tel. No ....

FARMING

Wairarapa farm software

Agrisoft Limited, a software house in Carterton, is a group of people with experience in farming, farm consultancy, and programming. They have written (and are writing) programs for several makes of computer. These include Peach, Microbee, TRS-80, System-80, Poly and MPF II (an Apple compatible machine).

The programs available at present are a budgeting suite, and a feed budget program. A herd recording system for TRS-80 and System 80 should be ready in June.

The budgeting suite of programs includes an annual budget, a monthly cash flow, a cash book, and a personal assets statement. Accompanying the programs is a very detailed manual.

The feed budget program is for cassette based machines and assesses feed reserves, predicted growth, and predicted demand. A final figure is produced indicating how much feed is left at the end of the budgeting period. This final figure may be either a surplus or a deficit.

These programs are retailed from Viscount Electronics in Palmerston North, and from some of the computer retailers.

In later issues there will be reviews of agricultural software, so you can expect to see reviews of Agrisoft programs.

LETTERS

FORTHERS

Dear Sir,

In reply to N. R. Moir's letter about FORTH. Among other things, in your October issue, I am writing to encourage other "FORTHers" to come out of their caves and speak up.

I am in third form at Sacred Heart College, Auckland - where we have five Pegasus FORTH's, one TINY BASIC, and one Pascal (also Pegasus).

Our computer studies and French teacher Mr John O'Gorman, is writing a program which will hopefully run FORTH on it.

Several boys in my class are making amazing programs on the FORTH computers. I find Forth very easy to use, once you know the basics. And I am looking forward to other writers telling "BITS & BYTES" subscribers about their experiences with FORTH. I am also looking forward to articles on the fantastic Forth language in later issues of "BITS & BYTES".

So get into gear all you other Forth lovers and speak up!

Yours sincerely,

Martin Yates

MICRO NEWS

From page 3

provides music and sound to rival even some dedicated music synthesizers. It produces three independent voices each with a range of nine octaves.

A CP/M cartridge will be available as will a PET emulator cartridge allowing 64 users to run PET software.

The Commodore 64 is also said to have impressive graphics.

PUZZLES

Not as a hobby

By ROGER ALTENA

This is a review of a new 8080-based computer on the market. It is marketed by NCR and features up to 256K of RAM, two membrane-type keyboards and a 32 by 16 green screen.

Mass storage is two high-speed cassette drives and an optional floppy disk drive. Hard copy is obtained from two 40-column dot matrix printers.

Also included is a novel method of data input - a magnetic-card reader similar to that used on TI and HP hand-held calculators.

The resident operating system is called FOS and allows one copy of a program to control over a dozen terminals.

I was disappointed with the keyboards. The letters A to Z, the digits 0 to 9, and three special function keys are included on the main keyboard, along with six separate function keys. There is also tactile and audible feedback on each key depression. However, fast typing speeds can definitely not be reached.

Built into the computer are several exotic hardware devices which will probably never be used by the average personal computer user. Another problem is the difficulty of developing programs written in a high-level language.

At present, the only high-level language available (called ATM/TPL) must be compiled on another machine, linked and dumped to cassette. This must then be loaded into the ATM where it can be run. This can be quite a nuisance for the average home computer owner.

Wrapping up, I found the ATM to be rather unsuitable for a hobbyist. Main problems include the size and weight (half a ton, and requiring over two square metres of desk space). The ATM is probably outside most people's budgets.

However, it is quite an efficient multi-processor and will certainly make an effective party piece.

Recognise this computer? A picture of it is on page 40.

Working in machine code

By Gordon Findlay

Judging from the people who contact me as a result of these columns, there is a lot of interest, and also a lot of confusion, about programming or machine code. I am not surprised — machine code is the only way to do some things. But don't use code if BASIC will do as good a job. This month I will address some of these questions.

Changing Machine Sizes.

Very often a useful utility or part of an interesting game will be presented in the form of a machine code routine, to be POKEd into high memory. But changing from, say, a 16K machine to a 48K machine isn't just a matter of changing the addresses you POKE into. It is seldom that a machine-code routine of any length is RELOCATABLE. A relocatable program is one which will operate regardless of where it is in memory.

The sort of things which stop programs, or subroutines, being relocatable are CALL or JP statements (like GOSUB and GOTO statements in BASIC if you like), and operations involving the stack. While it is possible to write every program in a relocatable manner, it isn't easy, and the result is increases in space required, and time.

So what do you do in such a case? The only really satisfactory method is to reassemble the program at the new location. If you have a listing of the source code (in mnemonic form) this is straight-forward, even if a lot of typing may be involved! If you don't have it in source form, but just as say, data statements, you must first disassemble the object code, modify it to reflect the new location, and reassemble. The things to watch out for are addresses in the area of the original program. Change them to the corresponding addresses in the new location.

Once you have relocated a program, it may be poked into the new location as required. Don't forget that BASIC can't handle addresses larger than 32767. Locations above that are referenced by subtracting 65536 from the actual address, so the top of a 32K machine is at 49151, which BASIC calls 49151 — 65536, = −16385.

ROM calls

All machine code programmers should take heed of the enormous resource of useful routines to be found in the level II ROMS. These ROMs are full of very ingenious code, all tried and fully debugged. So why reinvent the wheel by writing similar routines yourself?

There are three advantages in using ROM calls as much as possible:

- They are well-written, tested routines.
- They save you lots of programming time.
- They save space in your program.

There are also some potential disadvantages:

- You must watch the effect on the Z-80 registers carefully.
- You must also watch the effect on the stack.
- An error may dump you back in BASIC, with an error message which has nothing to do with your program.

Some of the dangers can be avoided by switching, temporarily, to the alternate register set, as at NO point does level II use the alternate set. I am not sure about DOS in this respect — can anybody enlighten us?

Here are some of the most useful ROM calls — not all of them, just ones which I have found particularly useful; the names I have invented — they are arbitrary:

For each routine I give a name, the address to call, the purpose, and the register usage. If a register is not mentioned, then it is probably left intact by the call — but check just to make sure.

SCAN (002BH) — instantaneously scans the keyboard, returning the code of a key being pressed in the accumulator, or zero if no key is being pressed. Uses DE registers.

KEYIIN (0049H) — like SCAN, but waits until a key is pressed. Also uses DE.

INPUT (0361H) — inputs a string (in ASCII) from the keyboard. The string is stored in the buffer pointed to by 40A7H (so assign the pointer first), and HL is left pointing to the location before the start of the buffer. All registers are used.

CHROUT (0049H) — outputs the contents of the accumulator to the screen, as an ASCII character. Uses DE. There are many other output routines in ROM, but this is the one I have found most useful. It updates the cursor position, and handles all the control codes — e.g. 31 (decimal) to clear to the end of the screen, 30 (dec.) to clear to the end of the current line, and so on.

MSGOUT (28A7H) — outputs a message, or string, to the video, at the current cursor position, and updates the cursor. To use it, load HL with the address of the first character in the string, and call MSGOUT. The end of the string should be followed by a byte of zeros. Only the accumulator is used.

Note: I have had occasional difficulties with this routine with NEWDOS80. The DOS call at 4467H works in the same way — but see the manual for delimiting.

RECON (0212H) — turns the tape recorder on. The recorder concerned is selected by the contents of the accumulator — 0 for tape recorder (the in-board recorder for the System-80), 1 for the second recorder. All registers

are used.

RECOFF (01F8H) — turns the tape recorder off. Uses AF only.

SYNC (0296H) — Reads tape, bypassing the leader, until the sync byte is read. Only the accumulator is used.

LEADER (0287H) — writes tape leader, and sync byte. Uses A and B registers.

TOUT (0264H) — writes the byte in the accumulator to tape — provided that the tape recorder has previously been started of course! Only the A-register is used.

TIN (0235H) — Reads a byte from tape, placing the result in the accumulator. The other registers are preserved.

Note on tape routines: timing is automatic for tape input and output. This means that you need only be sure to call TOUT or TIN often enough to keep up with 500 baud — you cannot go too fast for the tape, but could go too slowly (mind you, 500 baud is slow! Don't forget to disable interrupts.

CLS (01C9H) — clears the screen, and homes the cursor. Only the accumulator is clobbered. This routine resets the cursor location appropriately.

CHAR64 (04C3H) — converts the display to 64 character mode.

CHAR32 (04F6H) — converts the display to 32 character mode.

DELAY (0060H) — a delay loop. All registers are used. Load a counter in BC, and call DELAY. This produces a delay of 14.6 x (BC) microseconds.

CMP16 (0A39H) — this routine compares the BC and DE register pairs. The result is left in the A-register, and the flags are set. The contents of DE and HL are treated as 2 signed integers, and if they are equal, the A-register is set to 0. If DE VV HL, the A-register is set to −1, but if DE V HL, the A-register is set to +1. The flags reflect the state of A. Apart from AF, the registers are all preserved.

This routine is very useful for comparing addresses and the like.

CMPUN (1C90H) — this routine is the same as CMP16, except that the register contents are treated as unsigned integers.

PRTHL (0FAFH) — prints the contents of HL on video in decimal, at the present cursor position, and updates the cursor. All registers are used.

MOVE (09D7H) — moves a block of memory from the address pointed at by DE to the address given by HL. The number of bytes moved is first placed in the B-register. All the registers are used. Sometimes this is more convenient than an LDIR or LDDR, as it will move in the correct direction (up or down) as is necessary.

FIXBAS (1AF8H) — this routine should be called just before re-entering BASIC if your program has altered anything in the BASIC program area. The routine, which uses all registers, is the primary ROM routine for program maintenance, and it checks all the pointers from one line to the next, and updates the beginning and end of program pointers in the communications area.

Re-entry points.: Jump to these re-entry points to get out of your program:

- level II basic — 06CCH, or 1A19H — it doesn't seem to matter.
- DOS — 402DH.

These are only a few of the routines in the ROMs which you might want to use — there are a number of books available which give many more. You will notice that I haven't mentioned any of the arithmetic routines — I never seem to need them, so I am not familiar with them. No doubt many readers have their own favourites — write and tell me, and if there is enough interest I will list some more.

One last word of warning. Some of the books around are not absolutely accurate — especially about register usage — so do your own checks! I think all the above are correct, and I have used them all successfully, but you will want to check them, too.

From page 19

several things on its side. Because it was already well-established, its "swim-with-a-crowd" rating was high. Some of the best brains in the world were falling over themselves to write software for it and, in computers per cake stall, the price was great.

Within a staggeringly short time, schools went from having one computer to having three, four and five. More and more started toying with the idea of setting up a room with perhaps twice that number, all within a year.

Then the BBC Micro arrived . . . smarter than the Apple . . . cheaper than the Poly. At the beginning of 1983, it looked like THE machine for a class set and thoughts of de facto standardisation with Apple started to fade even before the newcomers had worn out their third set of Apple games paddles.

However, the recent announcement of cut-price imitation Apples imported direct through a school purchasing co-operative is spinning things round again. At time of writing, the news is still spreading only slowly, but the change from using a New Zealand distributor to what amounts to international mail-order promises to make this a most interesting few months.

Schools have shown they can rais money to buy computers and many have decided the time for a big push is now! What happens in the rest of 1983 is anyone's guess. As long as schools have to find the money though, the money will do the talking.

Michael Wall

Opening up the Beeble Box

By Pip Forer

What is a Beeble Box? Beeble Box is a regular feature, an ongoing exploration of the BBC computer. It is Beeble because the Beeb (also known as Auntie or the BBC) initiated its production and ensured its success. It is box because every computer, in whoever's hands, is a Pandora's box. Finally it is Beeble Box together in honour of the galactic hero, Zaphod. As you may recall he supported two heads : much of the attraction of the BBC computer is its two heads in the form of its main and second processors. But more of that later. This intorudction sets the scene for the upcoming television series, "The Computer Programme," and gives a little of the personality behind the BBC machine. I hope over the next few issues to evaluate some of its strengths (and weakenesses) but for now here is a brief summary of the history to date.

The BBC computer is argued by some (but by no means all) to be the current high point of low-cost educational/home computing using an 8-bit processor. It employs a 6502 processor and a very large (32K) operating system to drive an unusual but distinguished semi-structured BASIC. It has, in its slightly dearer version, a host of interfaces to the outside world, and thence eventually to other peripherals and cards. It is claimed to be networkable through a cheap but fast system ideal for class use in education. As Chris O'Donoghue and I agreed in the November, 1982, "Bits and Bytes," it is a machine of note as much for where it points as for what it is. Through its links with the BBC it has been guided towards a design that shuns the crassly commercial and been provided with publicity coverage that must make other manufacturers green with envy. Some have claimed that, like the St James translation of the Bible, it is a rare example of the successful design work of a committee. Technical comments on the equipment so far released have been positive with a few minor quibbles.

Where did this beast spring from? Its genesis goes back to the start of the 1980s when the BBC were seeking a machine to accompany a planned series of eductional programmes on computing. The specifications for the machine arose largely from the needs of its role as a teaching ambassador for computing. Any machine designed for this role had to be simple enough to catch the novice, open-ended enough to lead on to wider potential and serious enough to encourage good computer skills. When specifications were called it was widely believed that established micro makers (quite possibly the pyrotechnic Clive Sinclair) would get the contract. At this point enter a small company called Acorn Computers based, like Clive Sinclair, in Cambridge. From nowhere specifications were promised and, with much midnight oil and some disbelief, met. Exit one Clive Sinclair in dudgeon to design the Spectrum.

The BBC machine, as it was to become, bore similarities to an earlier Acorn machine, the Atom, which had already proved popular for its cheapness and graphics. The BBC machine began to appear in public at exhibitions in the Northern hemisphere Autumn of 1981 (although these models turned out to be cases connected to rather more substantial processors elsewhere). However, by the launch of the BBC programme that sired the machine (January, 1982) some machines were available for sale. Various firms became involved in their production, including the giants, ICL and Ferranti, who produced the special graphics chip in the machine. Larger numbers of units began to appear. However, due to production, and particularly, distribution problems, a substantial back-log of machine orders developed and the BBC ran "The Computer Programme" at off peak times first time round to keep demand down. It took until October, 1982, for this demand to dissipate. Disk drives are now available in the United Kingdom and production of second processors has begun. New languages are reputedly near release. Meanwhile the sneaky Australian educationists had got in on the act and siphoned machines off for development from mid-1982. Notwithstanding this we still get the programme (and wider public sale of the machine) before Australia.

With more than 70,000 units now delivered (a figure inevitably out of date by the time you read this) the initial delivery lags are behind the machine and a new production facility in Hong Kong should ensure no new lags develop. The BBC computer appears to be a major success story of British microcomputing.

What of Acorn computers? They continue to produce the Atom, and are developing a new machine called the Electron, and a high powered new processor based on the National 16032 chip (the Glueon). They also are very active in the fast-growing software industry associated with the BBC machine. Their executives, such as Chris Curry, plan major assaults on the international microcomputer markets (independently and through the BBC machine) and indulge in debate with their critics.

Yes, in this hullaballoo and general acceptance there are still critics. They include such models of modesty as Steve Jobs of Apple ("The BBC what?"), who may well believe his Macintosh will reduce all competitors to obsolescence, and Clive Sinclair (who suspects that his Spectrum already does). In a more unbiased vein several commentators have already bemoaned the low amount of free RAM once the operating system, graphics and other requirements have been met. Others comment on the early software deficiencies or the lack of alternative languages. The promise looks good for fast solutions to these problems, but there's many a slip twixt cup and lip.

The club man of Rangiora

Home computer enthusiasts from Kaitaia to Invercargill have joined the software club in Rangiora of Mr Dean Craw, the N.Z. Computer Games Club.

When interviewed recently, he had 230 members — and inquiries were coming in at the rate of three or four a day.

Users of TRS80/System 80, with about 38 per cent of the membership, lead in the brand breakdown, followed by Atari (about 19 per cent), then VIC 20, Apple, Commodore PET and Sinclair.

The TRS80/System 80 brigade love Tanktics, Defence Command, Star Fighter, and the Big Five series among the 150 or so games the club has available for the machines.

Mr Craw was an engineer in Rangiora with a passion for computer games when he set up his business last August. He has a shop, featuring home computers, video equipment, and games machines. He has the local market to himself, and draws visitors from Christchurch. But the club has reach. For example, he has about 40 members in the greater Auckland area.

He envisaged catering for isolated enthusiasts, and has been surprised by the interest from cities.

The club works this way. There is a $20 joining charge, and applicants nominate their machine section. They then get a newsletter outlining the games available.

A sliding hire charge is made for each game. For those with a retail price of up to $40, this is $4 for the first week and $2 for succeeding weeks. This ranges up to games listed between $100 and $120, which rent for $8 in the first week and $5 each succeeding week.

If the hirer damages the game, he must pay the full cost. If he wishes to buy it, he gets it at a

Dean Craw

discounted price which with the hire cost, brings it up to about retail price.

Mr Craw knows that one of the greatest problems of all software dealers is piracy, and one of the membership conditions is that rented games will not be copied. One member who disabled a tape by triggering an anti-copy guard has been expelled, says Mr Craw, and he promises the same fate to other pirates.

He feels that Video games machines are just getting off the ground, and has a good range for Atari and Fountain games systems. He also is gearing up for the BBC, and will be stocking the Dick Smith Wizzard, the Orbit Compu and the Tunix.

He has many educational programs, and plans to stock up on the Canterbury-developed farm and business software for the VIC', but won't be changing the emphasis from his first love — games.

And the favourite game of this connoisseur of electronic fantasy? Miner 2049ER, a new Atari game, is his present front runner. He believes it will become a classic.

GAMES

Builder

By JOHN BOWATER

This is a simple game, working on the basis of a "blob" which moves and leaves a trail on the screen.

It was created on a Commodore 4016, but should be easily adapted to run on other computers. The main changes would be in the "POKE" statements. Here is an explanation of what they do:

- "POKE 59468, 12" makes sure the characters are in uppercase.
- Lines 18 and 19 draw a border round the screen.
- Line 20 places some random obstacles on the screen.
- "POKE MC" on line 30 draws the blob itself. 'C' is the value of the character (a square), and 'M' is its position on the screen (top left of screen is 32768, bottom right is 33767).

It would be easiest to leave out the "PRINT SCORE" on line 20, all of line 21, the "PRINT T" on line 28, and all of line 29. These parts simply tell you your score during the game.

```
1 REM===================================
2 REM*********BUILDER*********
3 REM*****BY JOHN BOWATER****
4 REM*(THAMES, NEW ZEALAND)*
5 REM*COPYRIGHT 1982
6 REM===================================
7 POKE59468,12:CLR:PRINT"";TAB(16)"BUILDER
8 PRINTTAB(14)"__________
9 PRINT"   THE OBJECT OF THE GAME IS TO FILL
10 PRINT"   AS MUCH OF THE SCREEN AS YOU CAN,
11 PRINT"WITHOUT HITTING A WALL OR YOUR TRAIL.
12 PRINT"   YOU DO THIS BY DRIVING A CAR AROUND,
13 PRINT"USING THE KEYS AROUND '5' TO STEER IT.
14 PRINT"   CHOOSE SPEED:  1-(FAST)
15 PRINTTAB(13)"  15-(SLOW)"
16 INPUTR$:R=VAL(R$):IFR<1ORR>15THENRUN
17 R=R*42:C=102:Q=1:M=32767:PRINT""
18 FORI=1TO40:POKE32767+I,C:POKE33727+I,C:NEXT
19 FORI=1TO23:POKE32768+I*40,C:POKE32807+I*40,C:NEXT
20 FORI=1TO7:POKE32809+RND(1)*919,C:NEXT:PRINT"SCORE="
21 PRINT"";
22 REM:GAME ROUTINE=====================
23 GETZ$:Z=VAL(Z$):IFZ=2THENQ=40
24 IFZ=4THENQ=-1
25 IFZ=6THENQ=1
26 IFZ=8THENQ=-40
27 M=M+Q:IFPEEK(M)=CTHEN33
28 T=INT(S/845*100):PRINTT;
29 FORI=1TOLEN(STR$(T))+1:PRINT"";:NEXT
30 FORI=1TOR:NEXT:POKEM,C:S=S+1:GOTO23
31 REM:END==============================
32 FORJ=1TO40:NEXTJ:RETURN
33 FORI=1TO8:GOSUB32:PRINT"************";:GOSUB32
34 PRINT"            ";:GETZ$:NEXT:PRINT"AMOUNT COVERED=";T"%"
35 INPUT"PLAY AGAIN (Y/N)";B$
36 IFB$="Y"THENCLR:PRINT"":GOTO14
37 END
READY.
```


BOOKS

A good Apple manual from N.Z.

"Applesoft BASIC. A Teach-yourself Introduction," by Barrie M. Peake. Published by John McIndoe, of Dunedin. 118 pages. $9.95. Reviewed by Mike Wall.

The four meaty little books which come with every Apple computer cause all sorts of hassles. I've yet to find an Apple in use in New Zealand which doesn't have a disk drive with it but, because the disk drive is sold as an option, the books on programming make no mention whatsoever about using a disk.

The Applesoft Tutorial is definitely wordy, the Applesoft Reference Manual can tend to be terse while the Technical Reference Manual is written in hex. Users with the time eventually find their way round the different books, picking bits from here and there, but with any more than two or three people using the computer, this situation becomes impossible.

It is with this difficulty in mind, that Barrie Peake has written an "edited highlights" of the Apple literature. The strengths of the book are its price, its layout, its model answers, and the fact that you can get quite a wide view of the Apple system with one book instead of three or four.

Barrie Peake has been involved for some years in the teaching of high-level languages to university students. He is, by trade, a lecturer in chemistry (at the University of Otago) and with this background it is only natural to expect a numerical or "number-crunching" bias. Many people currently teaching programming who also lived through the FORTRAN era will feel instantly at home. The exercises at the end of each chapter, let the user explore a world where quadratics reveal

their roots, and every arithmetic progression has an nth term.

This "number" orientation is the only weakness in the book. With possible changes to the 7th form syllabus just around the corner, it seems a shame that strings and graphics are mentioned so briefly. I would have reservations about using this book with more junior classes because the language is too difficult and the approach is too "old-fashioned" for want of a better description.

Layout is interesting and, as all the computer programs contained in the book are printed straight from actual listings, I would expect the accuracy to be excellent. The many model answers are also going to be available on disk which is a great idea.

Design projects

"Z80 Microcomputer Design Projects." By William Barden, Jun. Reviewed by P. C. J. Parsonage. Published by Howard W. Sams & Co., Inc., U.S.A., 1981.

Have you ever wished you could build a circuit using a Z80 microprocessor to do something other than manipulate data on a television screen? Then this is the book for you.

You need very little background knowledge as the book is very clear. The first few chapters explain the operation of the various sections required to make a microcomputer. CPU, memory, and input/output are all covered briefly but adequately.

Programming instructions are explained in sufficient detail for the beginner to get started.

The Z80 microcomputer design described is very simple and full details, including printed circuit board design, are given. A circuit to programme EPROMs is required and that is also described with the same attention to detail.

Finally the author describes some applications. These include a burglar alarm, Morse-code generator, timer, music synthesiser, etc.

After getting a few of these working the reader should have no trouble designing his own applications.

This book is thoroughly recommended to anyone who has dabbled in electronics and feels it is time he built something using a microprocessor.

Good value for all

Computers for Everybody
By Jerry Willis and Merl Miller
dilithium Press - 1st edition
Reviewed by Cathy Arrow

An exciting and enlightening book. It introduces computers and imparts self confidence and knowledge to the reader.

Detailed steps in selecting a system for home or business, plus an excellent review of most popular computers provide a positive guide to the prospective purchaser. Prices in American dollars make Kiwis envious.

Uses for computers, types of programs, machine capabilities, and software applications give a solid foundation of essential information.

There are details of things you can add on and what they will do for you; plus what you could do with your computer right now.

"Where to find everything we didn't tell you," is the helpful conclusion to this book and describes the various magazines available.

"Computers for Everybody", is amusing, easy to read, assumes no previous computer or maths background, and is good value for "everybody" from teenager to octagenarian.

Additional features of the second edition are 16 colour pages and two new chapters, education and telecommunications. Business management and running a small business are expanded in the business chapter. Seventy computers are reviewed, and there are 101 things listed you could do with your computer right now.

BUSINESS

From page 16

From BASIC, text can be mixed with graphic screens in any specified character form, height and width. Commands are provided to set pen and fill colours, address dots in absolute or relative co-ordinates, position the graphics cursor, set viewports, and draw pre-defined images, along with many others. Images are stored in integer arrays, not shape tables!

Also provided with the system software is an emulation disk which allows the use of Apple II and II+ software. The only limitations are that the language card is not supported — so no Apple II Pascal, and only one version of BASIC is available at a time.

As we have come to expect from Apple, the documentation is impeccable. It is clear and easy to understand. The first manual, the "Apple III Owners Guide", takes the user step by step through setting up the machine, and operation. Don't be put off by the title of the second manual, "Standard Device Drivers Manual" — the title is the hardest bit.

Business BASIC documentation is also very good. It is not tutorial, but assumes some familiarity with computers and programming.

Software

For many applications, users will not need to write software: plenty is available. Indeed, many users probably haven't anything but commercially obtained programs.

All the old favourites are available in versions to make use of the III's capacity. These include Visicalc III, PFS III, PFS Report III, Desk Top Plan III, MicroModeller, accounting and payroll packages, mailing list management systems, word processing packages, and many others. While the III could be criticised for not running one of the standard operating systems, there is still plenty of software to choose from.

As well as all the usual text entry, editing, storing and printing features of such a package, Applewriter III will handle the automatic printing of form letters or other repetitive documents, automatically inserting names, addresses and other information from disk files which are built up using the text editor.

Complex word processing tasks can be carried out semi-automatically. Abbreviated forms can be entered into the text and will be automatically expanded before printing according to the glossary defined by the user. An extensive on-line "Help" file can be accessed at any time to get information about any particular feature. It is a very impressive piece of software.

Conclusion

The Apple III is a powerful, flexible synthesis of hardware and software. While some may criticise it for unadventurous architecture and for using a proprietary operating system, this does not apear to be a major disadvantage.

CLASSIFIEDS

FREE CLASSIFIED ADS

BITS & BYTES offers free classified advertisements of up to 20 words to members of micro clubs, students and hobbyists generally. Each word above 20 will cost 20c. If the advertisement is to appear more than once, then after the first insertion, the cost is 20c per word per issue.

VIC-20's for sale. For the best all-round deal ring Brian Bullen. Ph 298-8676 Papakura.

For Sale: VIC-20 Computers, peripherals and software. Discount prices and free software with every computer and peripheral. PERSONAL COMPUTER SYSTEMS, P.O. Box 860, Nelson. Telephone NN 79-362.

Wanted to buy: Applesoft Adventures on disk. No "all graphics" progams, and no machine language. John Bowater, 1 Oakley Cres., Thames.

For Sale: 48K plug-in memory board for TRS-80 complete with Rams, $200. Contact Briggs. 580 Lowe Street Hastings. Phone 69-808.

Statistics and Maths Programs Wanted: I am trying to collect together, for use by schools, any statistics or other Mathematics public domain programs which may be useful. Programs will be made available to all at only copying costs. If you have any which might be useful, please get in touch. Programs for any machine are acceptable. Please contact: Gordon Findlay, Head of Mathematics, Riccarton High School, Curletts Road, CHRISTCHURCH, 4.

For Sale: Adventure Games for Vic 20: Quest, Haunted House and Earthquake. $15 each or swap programs. Phone 478-138 Christchurch.

For Sale: Pegasus 6809 with, 4K Ram and other language on board. Contact Bill Quinn, 42 Rogers St., Blenheim. Phone 83-286.

For Sale: Microbee programs — Spreadsheet calculator; Othello, Fox and Hounds, graphics games etc. Write S. Doyle, 18 Holdsworth Ave, Upper Hutt.

For Sale: Business Computer, Canon CX1. With printer, good condition, under one year. Includes software. $9,200.00 ONO. Box 3985 AK, Ph 765-389 AK.

COMMERCIAL CLASSIFIEDS

Commercial Classifieds are available at 50c per word (minimum of 15 words). BITS & BYTES reserves the right to determine if any advertisement is a commercial classified. All commercial classifieds must be paid for before publication unless by prior arrangement.

CLUB CONTACTS

WHANGAREI COMPUTER GROUP: Tom Allan, 3 Maunu Rd, Whangarei. Phone 83-063 (w). Meets every second Wednesday of the month at Northland Community College.

NZ MICROCOMPUTER CLUB INC., P.O. Box 6210, Auckland. The Monthly Meeting is held the first Wednesday of each month at the VHF Clubrooms, Hazel Ave., Mt Roskill, from 7.30 p.m. Visitors are also welcome to the Computer workshop in the Clubrooms, 10am - 5pm, on the Saturday following the above meeting.

The following user groups are part of the club. All meetings shown start 7.30pm at the VHF Clubroom.

Other active User Groups within the club are:

APPLE, CP/M, DREAM 6800, SMALL BUSINESS, KIM, LNW, SORCERER, 1802 and 2650. They can all be contacted at club meetings or via NZ Microcomputer Club, P.O. Box 6210, Auckland.

APPLE USERS' GROUP: Bruce Given, 12 Irirangi Rd., One Tree Hill, Phone 667-720 (h).

ATARI MICROCOMPUTER USERS GROUP: Brian or Dean Yakas, Phone 8363 060 (h). Meetings: 1st Monday.

BIG BOARD USER GROUP: Steve Van Veen, Flat 5, 111 Melrose Rd, Mt Roskill, Auckland 4. Phone (09) 659-991 (h).

COMMODORE USERS' GROUP: Doug Miller, 18 Woldene Ave., Glenfield. Phone 444-9617 (h), 497-081 (w). Meetings: Third Wednesday.

CP/M USERS' GROUP: Kerry Koppert, 2/870 Dominion Rd., Balmoral. Phone 69-5355 (h).

DREAM 6800 USERS: Peter Whelan, 22 Kelston St, New Lynn, Auckland, Phone (09) 875-110 (h).

KIM USERS: John Hirst, 1A Northboro Rd, Takapuna, phone (09) 497-852 (h).

LNW USERS: Ray James, phone (09) 30-839 (w), 585-587 (h).

SINCLAIR USERS' GROUP: Doug Farmer, Phone 567-589 (h). Meetings: Fourth Wednesday.

SORCERER USERS' GROUP (NZ): Selwyn Arrow, phone 491-012 (h).

1802 USERS' GROUP: Brian Conquer, phone 655-984 (h).

2650 USERS' GROUP: Trevor Sheffield, phone 676-591 (h).

The above contacts can usually be found at NZ Microcomputer Club Meetings, or via P.O. Box 6210, Auckland.

Regular Meetings are:

MICRO CLUB, First Wednesday, plus an all day Computer Workshop the Saturday following. (10am – 5pm), all welcome.

ATARI MICROCOMPUTER USERS' GROUP: First Monday.

COMMODORE USERS' GROUP: Third Wednesday.

SINCLAIR USERS' GROUP: Fourth Wednesday. All meetings start at 7.30 pm at the VHF Clubrooms, at the end of Hazel Ave. (off Dominion Rd). Mt Roskill, Auckland.

Other Auckland-based groups:

ACES (Auckland Computer Education Society): Ray Clarke, 1 Dundas Pl., Henderson, Phone 836-9737 (h).

CMUG (Combined Microcomputer Users' Group): This is an association of Microcomputer Clubs, Groups, etc, formed to co-ordinate activities and to give a combined voice on topics concerning all micro users. Representation from all Clubs and Groups is welcomed to: CMUG C/- PO Box 6210, Auckland.

Epson HX20 Users' Group. Contact: C.W. Nighy, 14 Domett Avenue, Epsom, Auckland. (Ansaphone, 774-268).

HP41C USERS' GROUP (Auckland): C/- Calculator Centre, P.O. Box 6044, Auckland: Grant Buchanan, 790-328 (w). Meets third Wednesday, 7pm, at Centre Computers, Great South Rd., Epsom.

NZ TRS-80 MICROCOMPUTER CLUB: Olaf Skarsholt, 203A Godley Rd., Titirangi. 817-8698 (h). Meets first Tuesday, VHF Clubrooms, Hazel Ave., Mt Roskill, Auckland.

OSI USERS' GROUP (Ak): Vince Martin-Smith, 44 Murdoch Rd., Grey Lynn, Auckland. Meets third Tuesday, VHF Clubrooms, Hazel Ave., Mt Roskill.

SYMPOOL (NZ SYM USER GROUP): J. Robertson, P.O. Box 580, Manurewa, Phone 266-2188 (h).

A.Z.T.E.C.: Brian Mayo, Church Street, Katikati. Phone 490-326. Members use all micros and the club has just bought a Wizzard.

TAURANGA SINCLAIR COMPUTER CLUB: C. Ward, Secretary, P.O. Box 6037, Brookfield, Tauranga. Phone 82962 or 89234.

ATARI 400/800 USER CLUB: Dave Brown, P.O. Box 6053, Hamilton. Phone (071) 54-692 (h).

GISBORNE MICROPROCESSOR USERS' GROUP: Stuart Mullett-Merrick, P.O. Box 486, Gisborne, Phone 88-828.

ELECTRIC APPLE USERS' GROUP: Noel Bridgeman, P.O. Box 3105, Fitzroy, New Plymouth. Phone 80-216.

TARANAKI MICROCOMPUTER SOCIETY: P.O. Box 7003, Bell Block, New Plymouth: Francis Slater, Phone 84-514.

HAWKE'S BAY MICROCOMPUTER USERS' GROUP: Bob Brady, Pirimai Pharmacy, Pirimai Plaza, Napier, Phone 439-016.

MOTOROLA USER GROUP: Harry Wiggins, (ZL2BFR), P.O. Box, 1718, Palmerston North, Phone (063) 82-527 (h).

OSBORNE USER GROUP: Dr Jim Baltaxe, 18 Matipo St, Palmerston North, Phone (063) 64-411.

MICRO AND PEOPLE IN SOCIETY (MAPS): Levin. Meets on second and fourth Thursday of each month. Contacts: D. Cole, 28 Edinburgh St, Levin. Ph 83-904, or W. Withell, P.O. Box 405, Levin.

CENTRAL DISTRICTS COMPUTERS IN EDUCATION SOCIETY: Contact: Rory Butler, 4 John Street, Levin. (069) 84-466 or Margaret Morgan, 18 Standen Street,Karori, Wellington. (04) 767-167.

UPPER HUTT COMPUTER CLUB: Shane Doyle, 18 Holdworth Avenue, Upper Hutt. Phone 278-546. an all-machine club.

BBC USER GROUP. Users of other machines welcome too. Write P.O. Box 1581, Wellington, or Phone 861-213, Wellington.

NZ SUPER 80 USERS GROUP: C/- Peanut Computers, 5 Dundee Pl., Chartwell, Wellington 4. Phone 791-172.

Ohio Users' Group, Wellington. Secretary/Treasurer: R.N. Hislop, 658 Awatea Street, Porirua.

WELLINGTON MICROCOMPUTING SOCIETY Inc.: P.O. Box 1581, Wellington, or Bill Parkin (h) 725-086. Meetings are held in Wang's Building, 203-209 Willis Street, on the 2nd Tuesday each month at 7.30 p.m.

NELSON MICROCOMPUTER CLUB: Dr Chris Feltham, Marsden Valley Rd, Nelson. Phone (054) 73300 (h)

NELSON VIC USERS GROUP: Peter Archer, P.O. Box 860, Nelson. Phone (054) 79-362 (h).

BLENHEIM COMPUTER CLUB: Club night second Wednesday of month. Ivan Meynell, Secretary, P.O. Box 668. Phone (h) 86-207 or (w) 87-034.

CHRISTCHURCH ATARI USERS GROUP: Contact Edwin Brandt, 228-222 (H), 793-428 (W).

CHRISTCHURCH '80 USERS' GROUP: David Smith, P.O. Box 4118, Christchurch, Phone 63-111 (h).

CHRISTCHURCH PEGASUS USERS' GROUP: Don Smith, 53 Farquhars Rd, Redwood, Christchurch, Phone (03) 526-994 (h), 64-544 (w), ZL3AFP.

CHRISTCHURCH APPLE USERS' GROUP: Paul Neiderer, C/- P.O. Box 1472, Christchurch, Phone 796-100 (w).

OSI USERS GROUP (CH): Barry Long, 377 Barrington St., Spreydon, Christchurch. Phone 384-560 (h).

CHRISTCHURCH SINCLAIR USERS' GROUP: Contact, Mr J. Mitchell (385-141), P.O. Box 33 098.

CHRISTCHURCH COMMODORE USERS' GROUP: Contacts: John Kramer, 885-533 and John Sparrow, 896-099.

SOUTH CANTERBURY COMPUTERS' GROUP: Caters for all machines for ZX81 to IBM34. Geoff McCaughan, Phone Timaru 84-200 or P.O. Box 73.

LEADING EDGE HOME COMPUTER CLUB: Elaine Orr, Leading Edge Computers, P.O. Box 2260, Dunedin. Phone 55-268 (w).

NOTE: If your club or group is not listed, drop a line with the details to: Club Contacts, BITS & BYTES, Box 827, Christchurch. The deadline for additions and alterations is the second weekend of the month before the next issue.

GLOSSARY

Algorithm: A list of instructions for carrying out some process step by step.

Applications program: A program written to carry out a specific job, for example an accounting or word processing program.

BASIC: Beginners' All-purpose Symbolic Instruction Code. The most widely used, and easiest to learn, high level programming language (a language with English-like instructions) for microcomputers.

Binary: The system of counting in 1's and 0's used by all digital computers. The 1's and 0's are represented in the computer by electrical pulses, either on or off.

Bit: Binary digit. Each bit represents a character in a binary number, that is either a 1 or 0. The number 2 equals 10 in binary and is two bits.

Boot: To load the operating system into the computer from a disk or tape. Usually one of the first steps in preparing the computer for use.

Buffer: An area of memory used for temporary storage while transferring data to or from a peripheral such as a printer or a disk drive.

Bug: An error in a program.

Byte: Eight bits. A letter or number is usually represented in a computer by a series of eight bits called a byte and the computer handles these as one unit or "word".

Character: Letters, numbers, symbols and punctuation marks each of which has a specific meaning in programming languages.

Chip: An integrated circuit etched on a tiny piece of silicon. A number of integrated circuits are used in computers.

Computer language: Any group of letters, numbers, symbols and punctuation marks that enable a user to instruct or communicate with a computer. See also Programming languages and Machine language.

Courseware: Name for computer programs used in teaching applications.

CP/M: A disk operating system available for microcomputers using a particular microprocessor (that is the 8080 and Z80 based microcomputers such as the TRS 80 and System 80). See also Disk Operating Systems.

Cursor: A mark on a video that indicates where the next character will be shown, or where a change can next be made.

Disk: A flat, circular magnetic surface on which the computer can store and retrieve data and programs. A flexible or floppy disk is a single 8 inch or 5¼ inch disk of flexible plastic enclosed in an envelope. A hard disk is actually an assembly of several discs of hard plastic material, mounted one above another on the same spindle. The hard disk holds much more information – up to hundreds of millions of bytes – while floppy disks typically hold between 140,000 and three million bytes.

Disk drive: The mechanical device which rotates the disk and positions the read/write head so information can be retrieved or sent to the disk by the computer.

Diskette: Another name for a 5¼ inch floppy disk.

Disk operating system: A set of programs that operate and control one or more disk drives. See CP/M for one example. Other examples are TRSDOS (on TRS 80) and DOS 3.3 (for Apples).

DOS: See Disk Operating System.

Dump: Popular term for sending data from a computer to a mass storage device such as disks or tape.

Execute: A command that tells a computer to carry out a user's instructions or program.

File: A continuous collection of characters (or bytes) that the user considers a unit (for example an accounts receivable file), stored on a tape or disk for later use.

Firmware: Programs fixed in a computer's ROM (Read Only Memory); as compared to software, programs held outside the computer.

Floppy disks: See Disks.

Hard disks: See Disks.

Hardware: The computer itself and peripheral machines for storing, reading in and printing out information.

High-level language: Any Englishlike language, such as BASIC, that provides easier use for untrained programmers. There are now many such languages and dialects of the same language (for example MicroBASIC, PolyBASIC etc).

Input: Any kind of information that one enters into a computer.

Input device: Any machine that enters information into a computer. Usually done through a typewriter like keyboard.

Interactive: Refers to the "conversation" or communication between a computer and the operator.

Interface: Any hardware/software system that links a microcomputer and any other device.

I/O Acronym for "input/output".

KILOBYTE (or K): Represents 1024 bytes. For example 5K is 5120 bytes (5 x 1024).

Machine language: The binary code language that a computer can directly "understand".

Mass storage: A place in which large amounts of information are stored, such as a cassette tape or floppy disk.

Megabyte (or Mb): Represents a million bytes.

Memory: The part of the microcomputer that stores information and instructions. Each piece of information or instruction has a unique location assigned to it within a memory. There is internal memory inside the microcomputer itself, and external memory stored on a peripheral device such as disks or tape.

Memory capacity: Amount of available storage space, in Kbytes.

Menu: List of options within a program that allows the operator to choose which part to interact with (see Interactive). The options are displayed on a screen and the operator chooses one. Menus allow user to easily and quickly set into programs without knowing any technical methods.

Microcomputer: A small computer based on a microprocessor.

Microprocessor: The central processing unit or "intelligent" part of a microcomputer. It is contained on a single chip of silicon and controls all the functions and calculations.

Modem: Modulator-demodulator. An instrument that connects a microcomputer to a telephone and allows it to communicate with another computer over the telephone lines.

Program: A set or collection of instructions written in a particular programming language that causes a computer to carry out or execute a given operation.

RAM: Random access memory. Any memory into which you "read" or call up data, or "write" or enter information and instructions.

REM statement: A remark statement in BASIC. It serves as a memo to programmers, and plays no part in the running program.

Resolution: A measure of the number of points (pixels) on a computer screen.

ROM: Read only memory. Any memory in which information or instructions have been permanently fixed.

Simulation: Creation of a mathematical model on computers that reflects a realistic system.

Software: Any programs used to operate a computer.

Storage: See Mass storage.

System: A collection of hardware and software where the whole is greater that the sum of the parts.

Tape: Cassette tape used for the storage of information and instructions (not music).

VDU: Visual display unit. A device that shows computer output on a television screen.

Word: A group of bits that are processed together by the computer. Most microcomputers use eight or 16 bit words.

Puzzle answer — an automatic teller machine.